

# Rekognisi Huruf Tulisan Tangan Menggunakan Convolutional Neural Network

Fadhel Rahmawan  
Jurusan Teknik Informatika  
Universitas Logistik dan Bisnis  
Internasional  
Bandung Indonesia  
fadhel.rahmawan27@gmail.com

Roni Habibi  
Jurusan Teknik Informatika  
Universitas Logistik dan  
Bisnis Internasional  
Bandung, Indonesia  
roni.habibi@ulbi.ac.id

M. Yusril Helmi Setyawan  
Jurusan Teknik Informatika  
Universitas Logistik dan Bisnis  
Internasional  
Bandung, Indonesia  
yusrilhelmi@ulbi.ac.id

**Abstract**—The development of technology in the field of computer vision in recent years with the application of Technological developments in the field of computer vision in recent years with the application of convolutional neural networks have shown sophisticated performance with a high level of accuracy, such as object detection. The problem in the world of computer vision that has been looking for a solution for a long time is object classification in general images. How to duplicate the human ability to understand images, so that computers can recognize objects in images like humans. Therefore, the need for deep learning is one branch of machine learning where the algorithm used is inspired by the workings of the human brain. Some people may be more familiar with Convolution Neural Network. CNN is used to recognize and classify patterns in handwriting. The network assumes that the input used is an image. The network has a special layer called the convolution layer. In this layer, the images are inserted according to the predefined filters. In this study, various combinations of CNN architectural designs were carried out such as the number of convolution layers, stride size, number of epochs, type of kernel size optimizer. The research data comes from the National Institute of Standards and Technology (NIST) database, then the data is divided into three, namely 60% training data, 20% validation and 20% testing. The results of this experiment produce a very good accuracy value using 2 convolution layers, 50 epochs, with Adam optimizer producing an accuracy value of 99.5% when testing the model. Then evaluate the model using the confusion matrix, assigning a high value with an average value of 100% accuracy, while for the average value of precision with a value of 100%, for an average recall value of 100%, and finally an average value of f1 score of 100%.

**Keywords:** Deep Learning, Convolution Neural Network, Recognition, Image, Letter, Confusion Matrix.

## I. PENDAHULUAN

Salah satu problem dalam computer vision yang sudah lama diupayakan solusinya adalah pengenalan objek citra secara umum. Bagaimana menduplikasi kemampuan manusia dalam mengenali informasi citra, agar komputer dapat mengenali objek pada citra seperti manusia [1]. Gagasan klasifikasi citra adalah untuk memberikan pembelajaran komputer dari huruf yang diproses pada citra untuk menghasilkan huruf yang mewakili citra tersebut [2]. Salah satu pendekatannya adalah dengan menerapkan deep learning sebagai salah satu metode pembelajaran. Deep learning adalah subbidang dari AI yang secara tradisional dapat memberikan kemampuan belajar komputer tanpa pemrograman [3]. Metode neural network merupakan salah satu pendekatan supervised learning yang sangat berkembang pesat. Hal ini karena kemampuan neural network yang sangat luar biasa. Faktor performa menjadi hal yang diutamakan dalam neural network [4]. Metode Deep Learning saat ini yang memiliki hasil sangat signifikan dalam mengenali citra adalah Convolutional Neural Network [5]. Dikarenakan CNN tersebut berusaha meniru seperti sistem pengenalan citra yang terdapat pada visual cortex manusia sehingga memiliki kemampuan dalam mengolah atau mengenali informasi citra [6]. Metode Convolutional Neural Network dikenalkan sejak 1969 oleh Hubel (Neocognitron), lalu pada 1980-an dilanjutkan oleh Yann LeCun. Metode CNN merupakan hasil development dari metode Neural Network. CNN dapat digunakan sebagai metode dalam pengenalan pada objek citra.. Secara garis besar CNN tidak jauh berbeda dengan neural network lainnya [7]. Variasi dari citra akan sangat mempengaruhi hasil dari pelatihan, dan hal tersebut merupakan masalah yang cukup sulit untuk diatasi. Sebagaimana untuk mengklasifikasi huruf tulisan tangan akan membuat banyak sekali variasi pada setiap citranya seperti perbedaan bentuk dan pola. Pola adalah entitas yang telah ditentukan

sebelumnya yang dapat diidentifikasi sesuai dengan karakteristiknya, dan karakteristik ini digunakan untuk membedakan suatu pola dari pola lainnya [8]. Berdasarkan hal-hal tersebut di atas, maka perlu digunakan Convolutional Neural Network (CNN) untuk mengenali pola-pola yang terdapat pada tulisan tangan. Jaringan mengasumsikan bahwa input yang digunakan adalah data citra. Jaringan memiliki lapisan utama yang disebut lapisan convolution. Pada lapisan ini, gambar input diproses sesuai dengan filter yang ditentukan. Masing-masing lapisan ini menciptakan pola yang terdiri dari banyak bagian citra yang lebih mudah untuk dikenali nanti. Teknologi ini dapat membangun realisasi fungsi pembelajaran gambar lebih efisien. Dengan menggunakan metode CNN diharapkan akan didapatkan hasil akurasi yang tinggi dengan berbagai kombinasi arsitektur CNN pada kedua mekanisme pengujian yang dilakukan. Fokus utama dari pekerjaan ini adalah untuk mengetahui kemampuan CNN dalam mengenali karakter dari dataset gambar dan akurasi pengenalan dengan pelatihan dan pengujian menggunakan dataset NIST.

## II. STATE OF THE ART

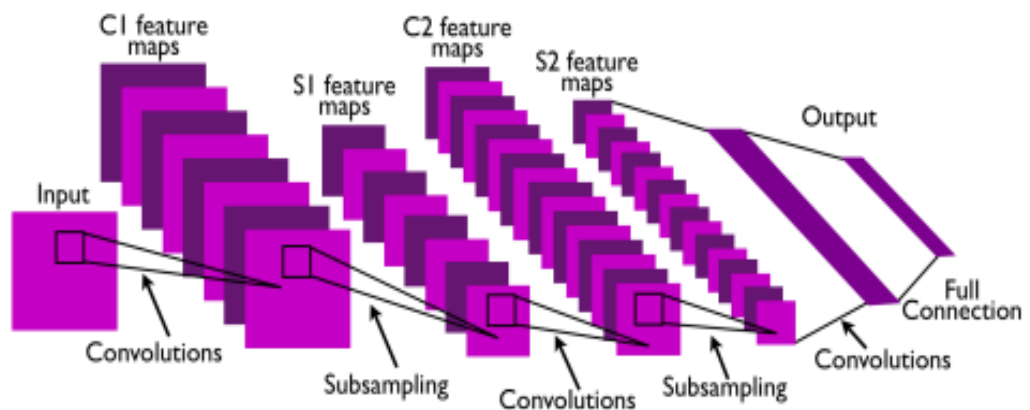
I Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir (2021) “Handwritten character recognition using convolutional neural network”, penelitian ini menggunakan dataset dari NIST (1200 data) yaitu data train 1000 dan data testing 200. Dengan membandingkan banyaknya data train dan mendapatkan hasil terbaik pada menggunakan 1000 data train dengan akurasi 92.91% [9].

## III. DATASET

Data yang digunakan pada penelitian ini yaitu sampel angka dari A-Z. Dataset huruf yang berasal dari National Institute of Standards and Technology (NIST) database. Berbentuk CSV (*Comma-Separated Values*), yang merupakan nilai dari tiap *pixel* gambar yang telah diconvert ke dalam bentuk matriks, data ini terdiri dari citra *grayscale* huruf dengan ukuran 28x28 *pixel*. Data yang dipakai dalam penelitian ini sebanyak 372.450 citra [10].

## IV. METODOLOGI PENELITIAN

Metode penelitian ini pembahasan tentang langkah yang dilakukan penulis pada penelitian. Dalam penelitian ini penulis menggunakan metode CNN untuk menyelesaikan penelitian. Tahapan yang dilakukan adalah sebagai berikut :



Gambar 1. Metodologi Penelitian

1. *Input*

*Input* adalah data citra yang digunakan untuk pengenalan data atau data latih.

2. *Convolutions*

*Convolution* adalah salah satu bagian dari proses *extrak fitur*, *convolution* dijalankan pada data input menggunakan sebuah *filter* atau *kernel* yang kemudian digunakan untuk memetakan fitur. Operasi *konvolusi* yaitu terjadi dengan menggeser filter di atas input. Di setiap lokasi, perkalian matriks dilakukan dan menjumlahkan hasilnya ke dalam peta fitur atau *feature maps*.

3. *Subsampling*

*Subsampling* merupakan bagian kedua dari ekstrak fitur setelah *convolution*, penggunaan metode *subsampling* dalam sebagian besar *cnn* adalah *max pooling* atau mengambil nilai *pixel* terbesar di setiap *pooling kernel*. maka, dengan mengurangi jumlah parameter sekalipun, informasi terpenting dari *pixel* tersebut tetap diambil.

4. *Fully connected layer*

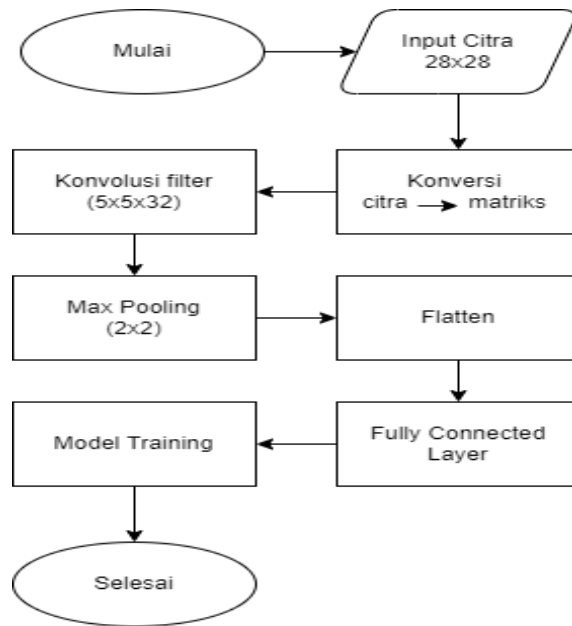
*Feature map* yang dihasilkan dari *feature extraction* masih berbentuk multidimensional array dan harus dilakukan konversi data dari fitur map menjadi array 1 dimensi (*flatten*). *Fully connected layer* terdiri dari banyak neuron, di mana masing-masing neuron terhubung ke semua neuron di lapisan selanjutnya. Lapisan-lapisan di ujung jaringan ini digunakan untuk membuat prediksi oleh jaringan dan untuk memberikan hasil non-linier akhir dari kombinasi fitur.

**a) Input Data Citra**

Input data citra adalah citra yang digunakan untuk data training/latih.

**b) Proses Training**

Proses *training* dilakukan untuk pengenalan bentuk-bentuk huruf menggunakan dataset. Pada proses *training*, pertama-tama membaca citra input berukuran 28x28, kemudian citra tersebut diubah kedalam bentuk matriks. Pada layer pertama terjadi proses konvolusi dengan 32 kernel berukuran 5x5 (5x5x32). Output dari hasil konvolusi akan digunakan sebagai input pada max pooling. Max pooling akan merubah ukuran citra, kemudian output yang terakhir pada proses *feature learning* akan diubah kedalam bentuk array 1 dimensi, proses tersebut dinamakan *flatten*. Kemudian dilakukan proses *feed forward* dan *backpropagation*. Selanjutnya prediksi dengan *fully connected layer* dan untuk mendapatkan nilai weight dan bias dari masing masing huruf A hingga Z digunakan fungsi aktivasi *softmax* sejumlah output yaitu 26.



Gambar 2. Proses Training pada CNN

**c) Model**

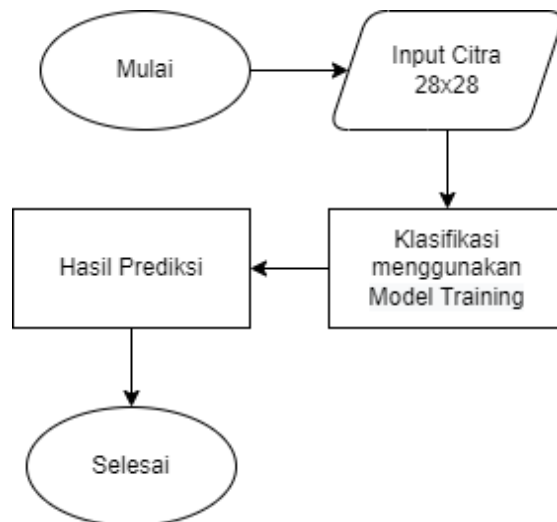
Merupakan hasil dari proses training yang nanti digunakan untuk proses *testing*.

**d) Input Data Testing**

Input data *testing* adalah citra yang akan digunakan pada proses *testing*.

**e) Identifikasi Menggunakan CNN**

Merupakan model yang dihasilkan dari proses *training* untuk digunakan dalam proses pengenalan citra.



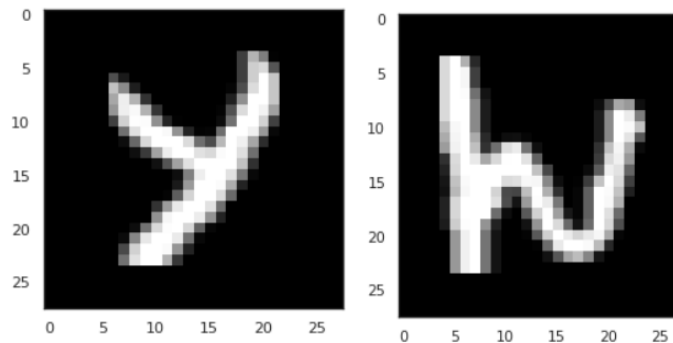
Gambar 3. Proses Testing

#### f) Hasil Pengenalan

Merupakan hasil akurasi identifikasi dari data citra menggunakan *CNN*.

### V. DATA YANG DIGUNAKAN

Data yang digunakan pada penelitian ini yaitu sampel angka dari A-Z. Dataset huruf yang berasal dari National Institute of Standards and Technology (NIST) database. Berbentuk CSV (*Comma-Separated Values*), yang merupakan nilai dari tiap *pixel* gambar yang telah diconvert ke dalam bentuk matriks, data ini terdiri dari citra *grayscale* huruf dengan ukuran  $28 \times 28$  *pixel*. Data yang dipakai dalam penelitian ini sebanyak 372.450 citra dan dibagi menjadi tiga kategori. Berikut tampilan data dibawah.



#### A Preprocessing

Data citra yang terdapat pada dataset tidak dapat langsung digunakan, maka dari itu diperlukan tahap preprocessing yang bertujuan untuk menghasilkan data citra latih yang lebih baik untuk diproses ke tahap selanjutnya. Tahapan preprocessing ini terdiri dari proses membagi data dan reshape.

##### a. Split/membagi Data

Membagi data jadi tiga kategori yaitu data *training*, *validasi* dan *testing*.

##### 1. Data Training

Data training merupakan data yang nantinya akan digunakan untuk melakukan proses *learning* huruf. Data yang digunakan untuk proses *training* sebanyak 223.470 citra untuk tujuh kategori.

##### 2. Data Validation

Data validation merupakan data yang akan digunakan untuk menguji tingkat akurasi saat melakukan proses *learning* pada tiap *epoch*. Data yang digunakan untuk proses *validation* sebanyak 74.490 citra.

##### 3. Data Testing

Data testing digunakan untuk menguji tingkat akurasi dari model yang sudah dilatih, data testing ini berbeda dengan data yang telah digunakan baik untuk proses *training* maupun proses *validation*. Dan jumlah data yang digunakan untuk *testing* berjumlah 74.490 citra.

##### b. Reshape

Data citra pada dataset masih berbentuk array satu dimensi yang mempunyai 2304 nilai, yang kemudian direshape ke bentuk array dua dimensi dengan nilai  $28 \times 28$ , dan mengubah tipe data menjadi *float* karena untuk proses normalisasi selanjutnya.

c. Normalisasi *Min-max*

Penggunaan satuan rentan nilai dapat mempengaruhi hasil dari analisis yang didapatkan, karena perbedaan nilai bobot yang diproses akan mempengaruhi hasil yang didapatkan, oleh karena itu dilakukan normalisasi *min-max*. Metode ini *re-scale* data dari suatu *range* ke *range* baru lain, nilai citra yang awalnya berkisar dari 0 sampai 255 diubah ke dalam *range* 0 sampai 1, dengan menggunakan persamaan 3.1 berikut:

$$x = \frac{x-min}{max-min} \quad (3.1)$$

## VI. PEMBANGUNAN MODEL

Secara global ada 3 konsep dasar pada *CNN* yaitu *Learning*, *validation*, dan *Testing*. Metode yang digunakan untuk mengembangkan aplikasi ini adalah *CNN* dengan arsitektur sistem seperti pada gambar dibawah.

Layer	Output Shape	Parameters
Input	28x28x1	0
Convolution + ReLU (1)	28x28x32	$(5 \times 5 \times 1 + 1) \times 32 = 832$
Max Pooling (1)	14x14x32	0
Dropout	14x14x32	0
Convolution + ReLU (2)	14x14x64	$(3 \times 3 \times 32 + 1) \times 64 = 18496$
Max Pooling (2)	7x7x64	0
Dropout	7x7x64	0
Convolution + ReLU (3)	7x7x128	$(3 \times 3 \times 64 + 1) \times 128 = 73856$
Max Pooling (3)	3x3x128	0
Dropout	3x3x128	0
Flatten	1152	0
Fully-Connected + ReLU (1)	256	$(1152 + 1) \times 256 = 295168$
Dropout	256	0
Output = Softmax	26	$(256 + 1) \times 26 = 6682$

Gambar 4. Arsitektur *CNN*

## VII. PEMBAHASAN DAN HASIL

Hasil dan pembahasan membahas hasil dari perancangan rekognisi huruf tulisan tangan yang meliputi pembahasan tentang bagaimana cara sistem bekerja dan hasil pengujian.

### a) Pengujian Arsitektur CNN

Untuk mengetahui program CNN sudah bekerja dengan baik, maka dilakukan pengujian pada arsitektur atau tiap layer-layer yang dibutuhkan untuk melatih model. Berikut beberapa layer yang diuji.

#### a. Convolution Layer

Pengujian *convolution layer* dilakukan dengan memberi inputan berupa sebuah citra grayscale ukuran 28x28 pixel. Kemudian dilakukan proses konvolusi dengan 32 kernel ukuran 5x5 atau berbentuk matriks dengan nilai `np.array([[ 1, 0, 0],[ 1, 1, 1], [ 1, -1, 1]])`, berikut blok program untuk *convolution layer*.

```
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
activation = 'relu', input_shape = (28,28,1)))
```

Setelah citra masuk ke program *convolution* maka akan didapatkan citra dengan dimensi yang lebih kecil yaitu 14x14 pixel, karena saat konvolusi tidak menggunakan padding.

#### b. Poling Layer

Pengujian *pooling layer* masih menggunakan nilai keluaran pada *convolution layer*, untuk pengujian *pooling layer* ini menggunakan kernel ukuran 2x2 dengan 2 *stride*, yang akan menghasilkan *feature map* setengah dari ukuran inputannya, sesuai dengan perhitungan *feature map*.

```
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
model.add(MaxPool2D(pool_size=(2,2)))
```

#### c. Flatten

*Flatten* digunakan untuk merubah bentuk dari *multidimensional array* menjadi *array* 1D atau *vector*, agar bisa digunakan pada *fully connected layer* dan dapat dihubungkan dengan *hidden layer*. Inputan dari *flatten* yang diuji adalah *array* yang sudah diproses di *pooling layer* berulang kali hingga dimensi dari *feature map* berukuran 6x6, berikut fungsi yang digunakan pada *flatten* untuk merubah bentuk dari *multidimensional array* ke *vector*.

```
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
model.add(Flatten())
```

#### d. Fully Connected Layer

Untuk pengujian *fully connected layer*, dengan cara membuat *hidden layer* yang akan dihubungkan dengan *vector* hasil *flatten* pada *layer* sebelumnya, selanjutnya melakukan operasi dot dari *layer* yang dibuat dengan *layer* sebelumnya dan diberikan fungsi aktivasi pada *layer* tersebut.

```

from keras.layers import Dense, Dropout, Flatten, Conv2D, Ma
xPool2D

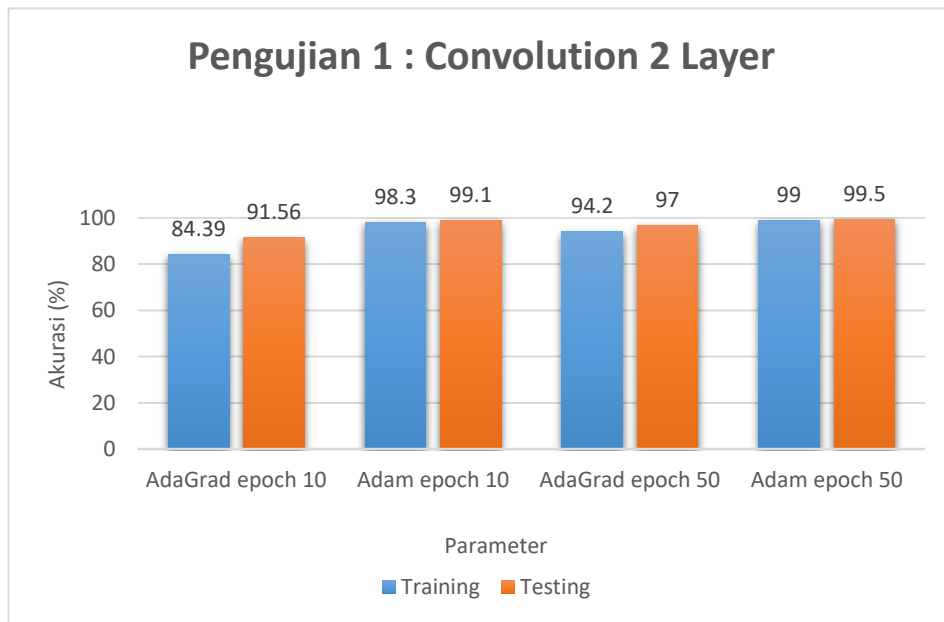
model.add(Dense(256, activation = "relu"))
model.add(Dense(26, activation = "softmax"))
    
```

**b) Proses Pengujian**

Pengujian dilakukan dengan menggunakan parameter yang telah ditentukan sebelumnya, untuk *optimizer* menggunakan dua jenis *optimizer*, yaitu Adam dan AdaGrad, untuk jumlah *convolution layer* adalah 2 lapisan dan 3 lapisan, untuk *batch size* yang digunakan adalah 128, dan yang terakhir perubahan nilai *epoch* yang digunakan adalah 10 dan 50. Untuk hasil pengujiannya dapat dilihat pada gambar berikut.

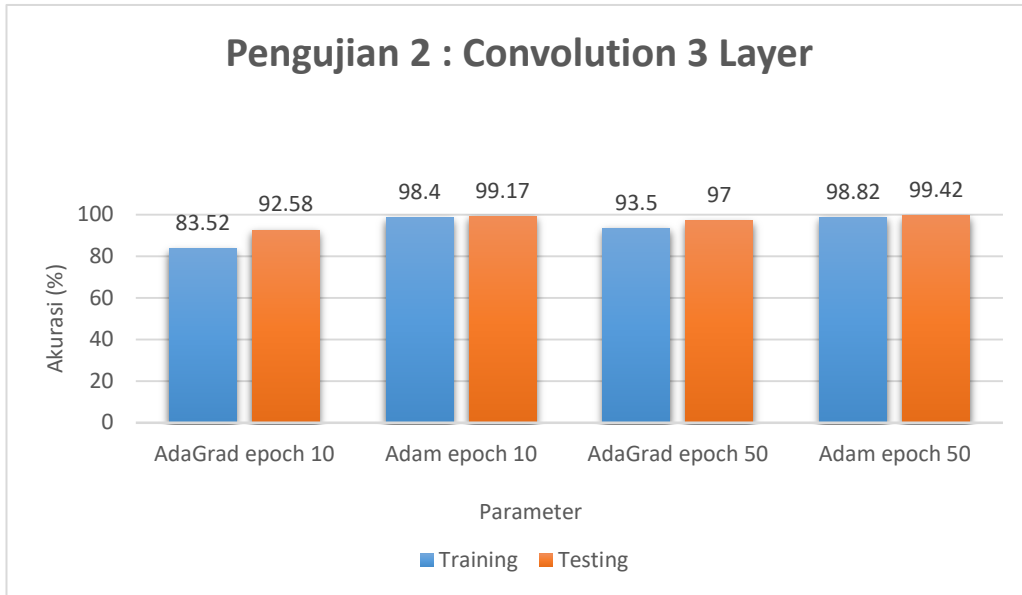
**a) Hasil Pengujian 1**

Berdasarkan hasil pengujian pertama seperti hasil yang ditunjukkan grafik hasil pelatihan yang menampilkan akurasi dari berbagai pemilihan parameter untuk pelatihan. Dapat dilihat pada grafik diatas, nilai akurasi tertinggi terdapat pada model yang menggunakan *optimizer adam* di *epoch* 50 dengan nilai sebesar 99% untuk *training* dan 99,5% untuk *testing*.



**b) Hasil Pengujian 2**

Berdasarkan hasil pengujian kedua seperti yang ditunjukkan grafik pengujian menggunakan 3 layer convolution, dan berbagai pemilihan parameter yang berbeda. Dapat dilihat nilai akurasi tertinggi berada pada model yang menggunakan *optimizer Adam* dengan banyak *epoch* 50 mendapatkan akurasi sebesar 98,82% untuk *training* dan 99,42% untuk *testing*.



Dapat dilihat pada kedua hasil pengujian bahwa nilai akurasi tertinggi diperoleh pada model yang menggunakan 2 layer convolution dan optimizer Adam pada epoch 50 dengan akurasi sebesar 99,5%. Bersumber dari <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning>, hal ini karena optimizer Adam merupakan pengembangan dari optimizer AdaGrad dan RMSProp. Dan semakin banyak jumlah epoch maka semakin tinggi juga tingkat akurasi pelatihan yang didapatkan.

**c) Evaluasi Model**

Pada evaluasi model menggunakan Confusion Matrix, evaluasi dilakukan dengan model paling unggul yang telah diuji sebelumnya, yaitu model yang menggunakan optimizer Adam dengan 2 layer konvolusi, activation function yang digunakan adalah ReLU pada tiap konvolusi layer termasuk pada hidden layer, serta softmax pada output layer, batch size sebesar 128 dengan epoch 50, dengan data train yang digunakan berjumlah 223.470. Sementara untuk data pengujian yang digunakan merupakan data yang terdapat pada dataset yang belum pernah digunakan pada proses pelatihan maupun proses validasi, sejumlah 74.490 citra yang terdiri dari huruf A-Z.

Evaluasi hasil klasifikasi dilakukan dengan menghitung nilai True Positive, True Negative, False Positive dan False Negative. Keempat nilai ini dapat dihitung berdasarkan confusion matrix seperti yang tertera pada Tabel 5.1 dibawah [11]. Nilai-nilai yang didapatkan dari confusion matrix selanjutnya digunakan untuk menghitung parameter-parameter evaluasi hasil klasifikasi.

Tabel 5.1 Confusion Matrix.

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Tabel 5.2 Confusion Matrix Hasil Pengujian Model.

		Prediksi																											
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
Aktual	A	2722	0	0	0	0	0	0	3	1	0	0	0	0	0	1	0	0	4	1	0	0	0	0	0	0	0	2	
	B	0	1630	0	2	0	0	0	0	0	0	0	0	0	0	4	2	0	0	4	0	0	0	0	0	0	0	0	1
	C	0	0	4707	0	3	0	5	0	0	0	3	2	0	0	1	0	1	1	6	0	2	0	0	0	0	0	0	0
	D	0	0	0	1900	0	0	0	0	0	3	0	0	0	0	11	7	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	1	1	0	2333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
	F	1	0	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	G	0	0	0	0	1	0	156	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	
	H	1	0	0	0	0	0	2	1437	0	1	2	0	1	4	0	2	0	0	0	0	0	0	0	1	0	0	0	0
	I	0	0	1	0	0	0	0	0	230	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	J	0	0	0	0	0	0	0	0	0	0	1651	0	2	0	0	0	0	0	0	0	4	3	0	0	0	0	0	0
	K	0	0	0	0	0	1	0	1	0	0	0	178	3	0	0	0	0	0	3	0	2	0	0	0	2	0	0	0
	L	1	0	13	0	0	0	0	0	0	0	0	2271	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2
	M	0	0	0	2	0	0	0	4	0	0	0	0	0	2489	4	3	0	0	0	0	1	0	0	0	0	0	0	0

N	0	0	0	0	0	0	0	0	2	0	0	0	0	0	3	3715	0	0	0	0	0	0	0	3	0	6	2	0	0	
O	0	2	2	45	0	0	3	0	0	0	0	0	0	0	0	11637	0	6	0	3	0	5	0	0	0	0	0	0		
P	0	1	0	9	0	2	0	0	0	0	0	0	0	0	0	13860	1	2	0	0	0	0	0	0	0	0	1	0		
Q	0	0	0	1	0	0	5	0	0	0	0	0	0	1	3	0	1146	0	0	0	0	0	0	0	0	0	2	0		
R	0	1	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	2280	0	0	0	0	0	0	0	0	0	0		
S	0	0	0	0	0	0	3	0	0	6	0	0	0	0	0	0	0	9655	0	0	0	0	0	0	0	0	0	0		
T	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	4501	0	0	0	0	0	0	0	0	3	1		
U	0	0	4	0	0	0	1	1	0	7	3	0	1	4	2	0	0	5751	2	9	0	2	0	0	0	0	0	0		
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5894	0	0	0	2	0	0	0	0	0	0		
W	0	0	0	0	0	0	0	0	0	2	0	0	1	4	0	0	0	0	2193	0	0	0	0	0	0	0	0	0		
X	0	0	0	0	0	0	0	1	0	0	4	0	0	0	0	0	0	1267	0	0	0	0	0	0	1	2	6	3	0	
Y	0	0	0	0	0	0	0	3	0	0	0	2	0	1	0	0	0	0	2098	0	0	0	0	1	4	0	1	2	6	0
Z	0	0	1	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1180	

Hasil klasifikasi dari model dengan citra sebanyak 74.490, yaitu. Huruf A terklasifikasi sebagai huruf A terdapat sebanyak 2722. Huruf B terklasifikasi sebagai huruf B sebanyak 1630. Huruf C terklasifikasi sebagai huruf C sebanyak 4707. Huruf D terklasifikasi sebagai huruf D sebanyak 1900. Huruf E terklasifikasi sebagai huruf E sebanyak 2333. Huruf G terklasifikasi sebagai huruf G sebanyak 1156. Huruf H terklasifikasi sebagai huruf H sebanyak 1437. Huruf I terklarifikasi sebagai huruf I sebanyak 230. Huruf J terklasifikasi sebagai huruf J sebanyak 1651. Huruf K terklasifikasi sebagai huruf K sebanyak 1178. Huruf L terklasifikasi sebagai huruf L sebanyak 2271. Huruf M terklasifikasi sebagai huruf M sebanyak 2489. Huruf N terklasifikasi sebagai huruf N sebanyak 3715. Huruf O terklasifikasi sebagai huruf O sebanyak 11637. Huruf P terklasifikasi sebagai huruf P sebanyak 3860. Huruf Q terklasifikasi sebagai huruf Q sebanyak 1146. Huruf R terklasifikasi sebagai huruf R sebanyak 2280. Huruf S terklasifikasi sebagai huruf S sebanyak 9655. Huruf T terklasifikasi sebagai huruf T sebanyak 4501. Huruf U terklasifikasi sebagai huruf U sebanyak 5751. Huruf V terklasifikasi sebagai huruf V sebanyak 894. Huruf W terklasifikasi sebagai huruf W sebanyak 2193. Huruf X terklasifikasi sebagai huruf X sebanyak 1267. Huruf Y terklasifikasi sebagai huruf Y sebanyak 2098. Huruf Z yang terklasifikasi sebagai huruf Z sebanyak 1180.

Dari tabel 5.2, dapat diperoleh nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Nilai TP, FP, FN dan TN dapat dilihat pada tabel 5.3 berikut.

Tabel 5.3 Nilai TP, FP, FN, TN dari huruf yang dikenali.

	TP	FP	FN	TN
A	2722	3	12	71752
B	1630	5	13	72841
C	4707	22	24	69736
D	1900	59	21	72509
E	2333	4	4	72148
F	240	3	1	73078
G	1156	19	3	73311
H	1437	17	14	73021
I	230	1	2	74256
J	1651	20	9	72809
K	1178	19	12	73280
L	2271	10	20	72188
M	2489	7	14	71979
N	3715	18	16	70740
O	11637	26	66	62760

	TP	FP	FN	TN
P	3860	11	17	70601
Q	1146	8	12	73323
R	2280	10	5	72194
S	9655	20	9	64805
T	4501	15	9	69964
U	5751	20	36	68682
V	894	3	7	73585
W	2193	18	9	72269
X	1267	10	9	73203
Y	2098	14	20	72357
Z	1180	6	4	73299
<b>Total</b>	<b>74121</b>	<b>386</b>	<b>368</b>	<b>1860690</b>

Dari tabel 5.3, dapat diperoleh nilai *accuracy*, *precision*, *recall* dan *f1 score* dari masing-masing huruf yang dikenali.

Untuk keseluruhan nilai *accuracy*, *precision*, dan *recall* pada tiap huruf dapat dilihat pada tabel 5.4 berikut.

Tabel 5. 4 Nilai *Accuracy*, *Precision*, *Recall*, *f1-score*, *support* dari masing-masing huruf yang dikenali.

	Precision	Recall	F1 Score
A	1.00	1.00	1.00
B	0.99	1.00	0.99
C	0.99	1.00	1.00
D	0.99	0.97	0.98
E	1.00	1.00	1.00
F	1.00	0.99	0.99
G	1.00	0.98	0.99
H	0.99	0.99	0.99
I	0.99	1.00	0.99
J	0.99	0.99	0.99
K	0.99	0.98	0.99
L	0.99	1.00	0.99
M	0.99	1.00	1.00
N	1.00	1.00	1.00
O	0.99	1.00	1.00
P	1.00	1.00	1.00
Q	0.99	0.99	0.99
R	1.00	1.00	1.00
S	1.00	1.00	1.00
T	1.00	1.00	1.00

	Precision	Recall	F1 Score
U	0.99	1.00	1.00
V	0.99	1.00	0.99
W	1.00	0.99	0.99
X	0.99	0.99	0.99
Y	0.99	0.99	0.99
Z	1.00	0.99	1.00
Akurasi			1.00
Nilai rata-rata bobot	1.00	1.00	1.00

Dapat dilihat dari table diatas nilai rata-rata *accuracy*, *precision*, *recall*, dan *f1score* yaitu :

*accuracy* = 1  
*recall* = 1  
*precision* = 1  
*F1 Score* = 1

## VIII. KESIMPULAN

Pemilihan parameter saat pelatihan antara lain, penentuan jumlah layer konvolusi, jumlah epoch, jenis optimizer maupun parameter lainnya dapat mempengaruhi akurasi dari model yang dilatih. Semakin banyak jumlah epoch maka semakin tinggi juga tingkat akurasi pelatihan yang didapatkan.

Dapat dilihat pada kedua hasil pengujian bahwa penelitian ini menghasilkan nilai akurasi yang sangat baik dengan menggunakan 2 layer konvolusi, 50 epoch, dengan optimizer adam menghasilkan nilai akurasi 99,5% pada saat pengujian model.

Nilai akurasi yang diukur pada tiap huruf yang diperoleh dari *confusion matrix* berkisar tinggi pada tiap huruf dengan nilai rata-rata akurasi sebesar 100%, sementara untuk nilai rata-rata presisi dengan nilai 100%, untuk nilai rata-rata *recall* sebesar 100%, dan terakhir nilai rata-rata *f1 score* sebesar 100%.

## IX. SARAN

Adapun yang dapat penulis sarankan sebagai pengembangan penelitian berikutnya sebagai berikut:

- a. Membangun tampilan antarmuka dengan desain yang user friendly atau yang lebih baik.

## REFERENCES

- [1] Feng, R., Gu, J., Qiao, Y & Dong, C., "Suppressing Model Overfitting for Image Super-Resolution Network," Computer Vision Foundation, June 2019, doi: <https://arxiv.org/abs/1906.04809>.
- [2] I Wayan Suartika E. P, Arya Yudhi Wijaya, and Rully Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *Jurnal Teknik ITS*, Vol 5, No 1 (2016), doi: <https://garuda.kemdikbud.go.id/documents/detail/1440423>.
- [3] Y. hen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "A Novel Deep Convolutional Neural Network For Spectral-Spatial Classification Of Hyperspectral Data," *Journal of Selected Topics in Developments, Technologies and Applications in Remote Sensing*, vol. 7, no. 6, 2018. Doi: 10.5194/isprs-archives-XLII-3-897-2018.
- [4] Siwi Prihatiningsih, Nadhiranisa Shafiy M, Feni Andriani, and Nurma Nugraha, "Analisa Performa Pengenalan Tulisan Tangan Angka Berdasarkan Jumlah Iterasi Menggunakan Metode Convolutional Neural Network," *Jurnal Ilmiah Teknologi dan*

- Rekayasa*, Vol. 24 No. 1, April 2019, doi: <https://garuda.kemdikbud.go.id/documents/detail/1364267>.
- [5] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, 36, 193-202 (1980), doi: <https://link.springer.com/article/10.1007/BF00344251>.
- [6] A.Karpathy, "CS231n Convolutional Neural Network for Visual Recognition," *Stanford University*, 2020, doi : <http://cs231n.stanford.edu/2020/>
- [7] Jiawei Han, and Micheline Kamber Jian Pei, "Data Mining: Concepts and Techniques. 3rd penyunt," *Morgan Kaufmann is an imprint of Elsevier*, MA 02451, 2012, doi: <http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>.
- [8] Sam'ani, and M. Haris Qamaruzzaman, "Pengenalan Huruf Dan Angka Tulisan Tangan Menggunakan Metode Convolution Neural Network (CNN)," *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, Volume 9, No 2 (2017), doi: <http://speed.web.id/ejournal/index.php/speed/article/view/296>.
- [9] I Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir, "Handwritten character recognition using convolutional neural network", *Journal of Physics: Conference Series*, Vol. 1918, 2021. Doi: <https://iopscience.iop.org/article/10.1088/1742-6596/1918/4/042152>.
- [10] NIST Special Database 19 "NIST Handprinted Forms and Characters Database" *STANDARD REFERENCE DATA*. Doi: <https://www.nist.gov/srd/nist-special-database-19>.
- [11] Damir Krstinić, Maja Braović, Ljiljana Šerić and Dunja Božić-Štulić, "Multi-Label Classifier Performance Evaluation With Confusion Matrix" *Artificial Intelligence and Machine Learning*, (2020). Doi: 10.5121/csit.2020.100801.
- [12] Najwa Altwaijry dan Isra Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Computing and Applications*. (2021). Doi: <https://doi.org/10.1007/s00521-020-05070-8>.
- [13] XenonStack, "Log Analytics With Deep Learning And Machine Learning". Medium, 13 Mei 2017. Doi: <https://medium.com/@xenonstack/loganalytics-with-deep-learning-and-machine-learning-20a1891ff70e>.
- [14] Hidayatullah, Priyanto, "Pengolahan Citra Digital Teori dan Aplikasinya," Penerbit Informatika Bandung. (2017). Doi: <https://opac.perpusnas.go.id/DetailOpac.aspx?id=1059250>.
- [15] Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab, Debdatta Kandar, "Handwritten Character Recognition from Images using CNN-ECOC", *Procedia Computer Science*, Vol. 167, 2020, Pages 2403-2409. Doi: <https://doi.org/10.1016/j.procs.2020.03.293>.