



Jurnal Politeknik Caltex Riau

Terbit Online pada laman <https://jurnal.pcr.ac.id/index.php/jkt/>

| e- ISSN : 2460-5255 (Online) | p- ISSN : 2443-4159 (Print) |

Perbandingan Arsitektur *LeNet* dan *AlexNet* Pada Metode *Convolutional Neural Network* Untuk Pengenalan *American Sign Language*

Muhammad Ezar Al Rivan¹ dan Alwyn Giovri Riyadi²

¹Universitas Multi Data Palembang, Teknik Informatika, email: meedzhar@mdp.ac.id

²Universitas Multi Data Palembang, Teknik Informatika, email: alwyngiovri99@gmail.com

Abstrak

American Sign Language (ASL) merupakan bahasa isyarat yang digunakan untuk berkomunikasi bagi penderita tuna rungu. Metode yang digunakan untuk mengenali ASL yaitu Convolutional Neural Network (CNN). Arsitektur yang digunakan LeNet dan AlexNet. Hasil dari masing-masing arsitektur kemudian dibandingkan kinerjanya. Penelitian dilakukan dengan 2 skema jumlah data yang digunakan, yaitu skema pertama 100 data per huruf dan skema kedua 1.000 data per huruf untuk menguji kinerja dari kedua arsitektur. Hasil penelitian setelah diuji dengan data baru, yaitu skema pertama untuk arsitektur LeNet menghasilkan akurasi keseluruhan 48,332% dan arsitektur AlexNet menghasilkan akurasi keseluruhan 32,584%. Skema kedua untuk arsitektur LeNet menghasilkan akurasi keseluruhan 92,468% dan arsitektur AlexNet menghasilkan akurasi keseluruhan 91,618%. Secara keseluruhan perbandingan dapat dikatakan bahwa arsitektur LeNet adalah arsitektur terbaik dalam penelitian ini.

Kata kunci: Klasifikasi, ASL, CNN, LeNet, AlexNet

Abstract

American Sign Language (ASL) is a sign language used to communicate for deaf people. The method used to identify ASL is Convolutional Neural Network (CNN). The architecture used by LeNet and AlexNet. The results of each architecture are then compared. The research was conducted with 2 schemes of the amount of data used, namely the first scheme of 100 data per letter and the second scheme of 1,000 data per letter to test the performance of the two architectures. The research results after being tested with new data, the first scheme for the LeNet architecture produces an overall accuracy of 48.332% and the AlexNet architecture produces an overall accuracy of 32.584%. The second scheme for the LeNet architecture produces an overall accuracy of 92.468% and the AlexNet architecture produces an overall accuracy of 91.618%. Overall comparison can be said that the LeNet architecture is the best architecture in this study.

Keywords: Classification, ASL, CNN, LeNet, AlexNet

1. Pendahuluan

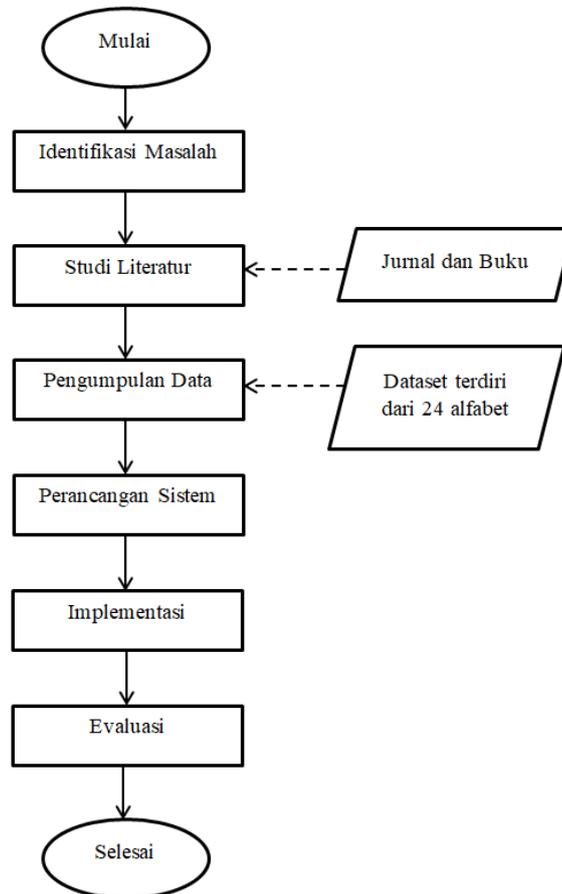
Bahasa isyarat adalah bahasa visual yang digunakan oleh penyandang tuna rungu dan tuna wicara untuk berkomunikasi dalam kegiatan percakapan sehari-hari [1]. Setiap negara memiliki bahasa isyarat yang berbeda-beda. Negara Amerika memiliki bahasa isyarat yang disebut *American Sign Language* (ASL) yang menjadi objek penelitian ini. ASL adalah bahasa isyarat yang bersifat linguistik yang sama dengan bahasa lisan yang diekspresikan dengan gerakan tangan. ASL banyak digunakan tuna rungu di Amerika Utara sebagai bahasa utama dalam berkomunikasi. ASL membantu tuna rungu untuk saling berkomunikasi dengan menggerakkan tangan dan membantu orang normal untuk memahami atau berkomunikasi dengan tuna rungu [2]. Terdapat beberapa penelitian mengenai ASL yaitu penelitian [3] menghasilkan nilai uji terbaik dengan hasil 0,99 saat menggunakan Euclidean Distance dan Manhattan Distance dengan $k = 3$ dan $k = 5$. Penelitian lain [4] berhasil melakukan pengenalan alfabet bahasa isyarat Amerika menggunakan *Edge Oriented Histogram* dan *Image Matching* yang menghasilkan kesimpulan bahwa fitur 9 bin dan metode *Sum of Absolute Difference* menjadi pilihan dalam pengenalan *American Sign Language*. Penelitian lainnya yang dilakukan [5] menghasilkan nilai rata-rata akurasi berdasarkan skenario 3 jumlah neuron yaitu 5 neuron sebesar 95,38%, 10 neuron sebesar 96,64%, dan 15 neuron sebesar 97,32%.

Pemanfaatan teknologi *machine learning* berupa metode *Convolutional Neural Network* (CNN) yang termasuk salah satu konsep dari *deep learning* yang mampu mengenali bentuk objek khususnya citra atau gambar, sudah berkembang dan mengalami kemajuan yang signifikan. Pemanfaatan CNN dapat digunakan untuk beberapa masalah seperti *image classification*, *object detection*, *object localization*, dan *image segmentation*. Penelitian yang telah dilakukan sebelumnya mengatakan bahwa tingkat keberhasilan dari metode CNN sangat baik dengan tingkat akurasi yang tinggi terhadap berbagai jenis citra yang diberikan. Penelitian yang dilakukan [6] menghasilkan akurasi lebih dari 89%. Penggunaan CNN pada penelitian ini akan dilakukan *tunning parameter* berdasarkan jumlah *feature maps* sampai hasil performa *training* yang dihasilkan stabil dan sesuai dengan yang diharapkan untuk mendapatkan nilai yang maksimal, seperti penelitian [7] yang memanfaatkan arsitektur *LeNet* dengan menggunakan *feature maps* berjumlah 32 *feature maps* pada *layer convolutional* pertama dan 64 *feature maps* pada *layer convolutional* kedua untuk mendapatkan hasil yang maksimal.

Metode CNN mengalami perkembangan dengan beberapa variasi arsitektur seperti *LeNet*, *AlexNet*, *VGGNet*, *GoogLeNet*, *ResNet*, dan lainnya. Arsitektur *LeNet* dan *AlexNet* merupakan arsitektur CNN yang memiliki rata-rata tingkat keberhasilan yang tinggi dari arsitektur lainnya. Arsitektur *LeNet* dan *AlexNet* pada penelitian yang dilakukan [8] menghasilkan perbandingan *Error Rate*, dengan *AlexNet* menghasilkan *Error Rate* paling besar 15,40% dengan data latih gambar paling besar yaitu 15 juta dan *LeNet* menghasilkan *Error Rate* paling kecil 1,10% dengan data latih gambar paling kecil yaitu 60 ribu. Perbandingan lainnya dari penelitian [9] menghasilkan akurasi *AlexNet* lebih tinggi dari *LeNet* dengan data MNIST dari *AlexNet* 99,2% dan *LeNet* 97,2%, untuk data IRIS dari *AlexNet* 98,6% dan *LeNet* 96,4%. Penelitian [7] menggunakan arsitektur *LeNet* menghasilkan akurasi tertinggi yaitu 98,89%. Selanjutnya penelitian [10] menggunakan arsitektur *LeNet* menghasilkan akurasi diatas 98%. Adapun penelitian [11] menggunakan arsitektur *AlexNet* mampu menghasilkan akurasi 85% hingga 90% berdasarkan data latih yang digunakan dengan kategori 20 kelas. Penelitian perbandingan *AlexNet* dengan arsitektur lainnya yang dilakukan [12] menghasilkan akurasi 98,93% dengan waktu proses paling cepat 94,17 menit dibandingkan arsitektur lainnya. Dengan adanya suatu sistem yang dapat mengenali pola alfabet dalam bahasa isyarat, maka dapat membantu masyarakat umum dalam berkomunikasi dengan bahasa isyarat

2. Metode Penelitian

Tahapan penelitian diawali dengan identifikasi masalah, studi literatur, pengumpulan data, perancangan sistem, implementasi dan evaluasi. Rancangan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1 Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah penelitian mengenai perbandingan arsitektur *LeNet* dan *AlexNet* pada metode CNN dan mengenai klasifikasi dari objek alfabet dari *American Sign Language*.

2.2 Studi Literatur

Pada tahap ini melakukan studi literatur dalam penelitian yaitu mengumpulkan data atau informasi yang diperoleh dari buku dan jurnal orang lain yang pernah dibuat sebelumnya serta dapat menjadi gambaran berapa perbedaan penelitian yang dilakukan.

2.3 Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data yang diperlukan berkaitan dengan penelitian yang akan dikerjakan yaitu gambar dari *American Sign Language* yang didapatkan dari [13] dan bersifat publik. Fitur yang digunakan dari citra berupa semua *pixel* yang ada pada gambar. Proses

ekstraksi fitur dilakukan oleh CNN. CNN juga dapat digunakan sebagai ekstraksi fitur selain digunakan sebagai klasifikasi. Penggunaan jumlah data pada penelitian dibagi menjadi 2 skema, untuk skema pertama menggunakan 100 gambar per kelas (total 2.400) dengan pembagian 80% data training, 10% data validation, dan 10% data testing. Pada skema kedua menggunakan 1.000 gambar per kelas (total 24.000) dengan pembagian 80% data training, 15% data validation, dan 5% data testing. Rincian skema penggunaan data dapat dilihat pada Tabel 1.

Tabel 1. Rincian Skema Penggunaan Dataset

No	Skema	Train	Validation	Testing	Jumlah
1	Pertama	1.920	240	240	2.400
2	Kedua	19.200	3.600	1.200	24.000

2.4 Pelatihan CNN

Pada tahapan ini, data latih digunakan pada CNN dengan arsitektur *LeNet* dan *AlexNet* agar dapat mengenali objek untuk dilakukan data pengujian. Proses pelatihan kedua arsitektur dilakukan dengan susunan *layer* dan parameter yang disesuaikan dengan performa *training* yang dapat dilihat pada Tabel 2 untuk arsitektur *LeNet* dan Tabel 3 untuk arsitektur *AlexNet*.

Tabel 2. Arsitektur LeNet

	Layer	Feature Map	Size	Kernel Size	Stride	Activation
<i>Input</i>	<i>Image</i>	3	32 x 32	-	-	-
1	<i>Convolution</i>	32	28 x 28	5 x 5	1	<i>ReLU</i>
2	<i>Max Pooling</i>	32	14 x 14	2 x 2	1	<i>ReLU</i>
3	<i>Convolution</i>	64	10 x 10	5 x 5	1	<i>ReLU</i>
4	<i>Max Pooling</i>	64	5 x 5	2 x 2	1	<i>ReLU</i>
5	<i>Convolution</i>	128	1 x 1	5 x 5	1	<i>ReLU</i>
6	FC	-	512	-	-	<i>ReLU</i>
<i>Output</i>	FC	-	24	-	-	<i>Softmax</i>

Tabel 3. Arsitektur AlexNet

	Layer	Feature Map	Size	Kernel Size	Stride	Activation
<i>Input</i>	<i>Image</i>	3	227 x 227	-	-	-
1	<i>Convolution</i>	32	55 x 55	11 x 11	4	<i>ReLU</i>
2	<i>Max Pooling</i>	32	27 x 27	3 x 3	2	<i>ReLU</i>
3	<i>Convolution</i>	64	23 x 23	5 x 5	1	<i>ReLU</i>
4	<i>Max Pooling</i>	64	11 x 11	3 x 3	2	<i>ReLU</i>
5	<i>Convolution</i>	96	9 x 9	3 x 3	1	<i>ReLU</i>
6	<i>Convolution</i>	96	7 x 7	3 x 3	1	<i>ReLU</i>
7	<i>Convolution</i>	64	5 x 5	3 x 3	1	<i>ReLU</i>
8	<i>Max Pooling</i>	64	2 x 2	3 x 3	2	<i>ReLU</i>

9	FC	-	64	-	-	<i>ReLU</i>
10	FC	-	64	-	-	<i>ReLU</i>
<i>Output</i>	FC	-	24	-	-	<i>Softmax</i>

2.5 Model

Pada tahapan ini dilakukan penyimpanan proses dari pelatihan data dalam bentuk model dengan format H5 atau disebut sebagai *Hierarchical Data Format* (HDF) yang merupakan sekumpulan format *file* (HDF4 dan HDF5) yang didesain untuk menyimpan dan mengatur data yang sangat besar dengan menggunakan bahasa pemrograman *Python*, kemudian model tersebut akan digunakan pada tahap proses pengujian menggunakan data uji. Setelah dilakukan pengujian pada data uji dari proses pengujian tersebut akan menghasilkan *confusion matrix*.

2.6 Analisis hasil

Tahap ini dilakukan untuk mengetahui tingkat akurasi keberhasilan dan perbandingan dari kedua arsitektur pada CNN pada program dalam mengklasifikasikan alfabet dari *American Sign Language*. Untuk mengetahui tingkat keberhasilan dapat dilakukan dengan menghitung tingkat akurasi, presisi dan recall. Adapun formulayang digunakan untuk menghitung keberhasilan klasifikasi seperti berikut :

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

$$presision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

Keterangan :

TP adalah *true positive* yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.

TN adalah *true negative* yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

FP adalah *false positive* yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

FN adalah *false negative* yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.

3. Hasil dan Pembahasan

3.1 Implementasi CNN

Pada tahap ini dilakukan implementasi metode CNN dengan arsitektur *LeNet* dan *AlexNet* sebanyak 5 kali untuk memperoleh nilai rata-rata *accuracy*, *precision*, dan *recall* dengan jumlah iterasi (*epoch*) berjumlah 10 pada skema pertama dan skema kedua penelitian.

3.2 Hasil Pengujian CNN

Pada tahap ini dilakukan pengujian model yang sudah disimpan menggunakan data uji berdasarkan skema penggunaan jumlah data. Pada skema pertama untuk pengujian per kelas dapat dilihat pada Tabel 4 dan skema kedua pada Tabel 5.

Tabel 4. Hasil Pengujian Per Kelas Skema Pertama

Arsitektur	<i>LeNet</i>			<i>AlexNet</i>		
Kelas	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
A	68	55	68	48	47,6	48
B	52	51,4	52	30	31,6	30
C	84	79,6	84	40	50,6	40
D	70	48	70	62	50,8	62
E	46	48,8	46	34	28,6	34
F	40	86,8	40	50	62	50
G	54	53	54	42	40,4	42
H	76	56,4	76	48	70,4	48
I	66	56,2	66	34	41,6	34
K	42	42,8	42	8	44,4	8
L	70	66	70	56	28	56
M	46	44,2	46	6	23,4	6
N	28	52	28	34	53,4	34
O	40	55,8	40	40	29,2	40
P	70	76,4	70	32	47,8	32
Q	72	74,6	72	52	49,2	52
R	36	39,6	36	22	22,8	22
S	16	20,8	16	16	38	16
T	36	46,6	36	26	31	26
U	38	41	38	8	15,6	8
V	16	21,8	16	20	13,4	20
W	22	20,6	22	16	23,2	16
X	36	28,2	36	16	16	16
Y	36	57,8	36	42	33	42

Pada Tabel 4 dapat diketahui untuk arsitektur *LeNet* bahwa nilai *Accuracy* tertinggi terdapat pada kelas C sebesar 84% dan *Accuracy* terendah pada kelas S sebesar 16%. Nilai *Precision* tertinggi terdapat pada kelas F dengan tingkat sensitivitas sebesar 86,6% dalam membedakan objek lain dan *Precision* terendah pada kelas W sebesar 20,6%. Nilai *Recall* tertinggi terdapat pada kelas C sebesar 84% dalam mengenali objeknya sendiri dan *Recall* terendah pada kelas S sebesar 16%. Pada arsitektur *AlexNet* dapat diketahui bahwa nilai *Accuracy* tertinggi terdapat pada kelas D sebesar 62% dan *Accuracy* terendah pada kelas M sebesar 6%. Nilai *Precision* tertinggi terdapat pada kelas H dengan tingkat sensitivitas sebesar 70,4% dalam membedakan objek lain dan *Precision* terendah pada kelas V sebesar 13,4%. Nilai *Recall* tertinggi terdapat pada kelas D sebesar 62% dalam mengenali objeknya sendiri dan *Recall* terendah pada kelas M sebesar 6%.

Tabel 5. Hasil Pengujian Per Kelas Skema Kedua

Arsitektur	<i>LeNet</i>			<i>AlexNet</i>		
	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
A	94	93,2	94	93,6	91,6	93,6
B	97,2	89,8	97,2	95,2	91,8	95,2
C	98,8	96,2	98,8	99,6	92,6	99,6
D	96	92,6	96	94,4	98	94,4
E	87,6	95,4	87,6	86,8	84,4	86,8
F	93,6	96,6	93,6	96,4	93	96,4
G	96,4	98,4	96,4	92,4	94,6	92,4
H	97,6	98	97,6	92,4	97,2	92,4
I	96,4	90	96,4	95,6	91,6	95,6
K	92,8	90,6	92,8	88	93	88
L	95,6	94	95,6	94	98,6	94
M	90,4	91,6	90,4	84,8	92,6	84,8
N	94	95,6	94	89,2	91	89,2
O	91,6	94,6	91,6	89,2	92,8	89,2
P	95,2	97,2	95,2	92	93,8	92
Q	94,4	96,8	94,4	96,8	94,6	96,8
R	85,2	86,8	85,2	87,2	89,4	87,2
S	87,2	97,2	87,2	84,8	85,2	84,8
T	92,4	96,8	92,4	95,2	91,6	95,2
U	88,4	79,6	88,4	87,6	83	87,6
V	80,4	87	80,4	84,8	88	84,8
W	89,2	87,4	89,2	92,8	97,8	92,8
X	88,4	90,2	88,4	90,8	87	90,8
Y	96,4	93,4	96,4	94,8	96,4	94,8

Pada Tabel 5 dapat diketahui untuk arsitektur *LeNet* bahwa nilai *Accuracy* tertinggi terdapat pada kelas C sebesar 98,8% dan *Accuracy* terendah pada kelas V sebesar 80,4%. Nilai *Precision* tertinggi terdapat pada kelas G dengan tingkat sensitivitas sebesar 98,4% dalam membedakan objek lain dan *Precision* terendah pada kelas U sebesar 79,6%. Nilai *Recall* tertinggi terdapat pada kelas C sebesar 98,8% dalam mengenali objeknya sendiri dan *Recall* terendah pada kelas V sebesar 80,4%. Pada arsitektur *AlexNet* dapat diketahui bahwa nilai *Accuracy* tertinggi terdapat pada kelas C sebesar 99,6% dan *Accuracy* terendah pada kelas S dan V sebesar 84,8%. Nilai *Precision* tertinggi terdapat pada kelas L dengan tingkat sensitivitas sebesar 98,6% dalam membedakan objek lain dan *Precision* terendah pada kelas U sebesar 83%. Nilai *Recall* tertinggi terdapat pada kelas C sebesar 99,6% dalam mengenali objeknya sendiri dan *Recall* terendah pada kelas S dan V sebesar 84,8%.

3.3 Hasil Perbandingan Kedua Arsitektur

Pada tahap ini merupakan hasil perbandingan dari arsitektur *LeNet* dan *AlexNet* dari hasil pelatihan dan pengujian data yang dapat dilihat pada Tabel 6.

Tabel 6. Perbandingan Arsitektur *LeNet* dan *AlexNet*

Arsitektur	<i>LeNet</i>		<i>AlexNet</i>	
	Pertama (100 Data)	Kedua (1000 Data)	Pertama (100 Data)	Kedua (1000 Data)
<i>Time/sec</i>	23,6	230	132,8	1.355,4
<i>Train loss</i>	0,6265	0,2081	1,8666	0,22282
<i>Train accuracy</i>	0,79438	0,93702	0,40958	0,93356
<i>Val loss</i>	2,39492	0,31036	2,3384	0,30774
<i>Val accuracy</i>	0,45804	0,9195	0,305	0,91684
<i>Accuracy Overall</i>	48,33%	92,47%	32,58%	91,62%
<i>Precision Overall</i>	50,98%	92,88%	37,17%	92,07%
<i>Recall Overall</i>	48,33%	92,47%	32,58%	91,60%
Tingkat Kenaikan Akurasi	44%		59%	
<i>Trainable Params</i>	336.984		278.872	
Kapasitas Penyimpanan	4,09 MB		3,42 MB	

Pada Tabel 6 dapat diketahui perbandingan masing-masing arsitektur yang menghasilkan kesimpulan bahwa secara keseluruhan bahwa arsitektur *LeNet* adalah arsitektur terbaik dalam penelitian ini, karena waktu proses pelatihan model lebih cepat dengan *trainable params* yang lebih banyak dan menghasilkan nilai akurasi pengujian data lebih tinggi di setiap pengujian skema pertama dan skema kedua, serta kemampuan dari setiap kelas untuk mengenali kelasnya sendiri yang tinggi berdasarkan nilai *Precision* dan *Recall*, sehingga menjadi arsitektur yang relevan dalam penelitian ini.

4. Kesimpulan dan Saten

4.1 Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa dalam pengenalan ASL, arsitektur *LeNet* merupakan arsitektur terbaik dibandingkan *AlexNet*. Arsitektur *LeNet* memiliki struktur yang lebih ringkas dibandingkan dengan arsitektur *AlexNet*. Arsitektur *LeNet* memiliki *trainable parameter* yang lebih banyak dibandingkan dengan arsitektur *AlexNet* sehingga *accuracy* pada *LeNet* lebih baik. Hasil secara keseluruhan dari skema pertama, *LeNet* menghasilkan nilai *accuracy* 48,33%, *precision* 50,98%, dan *recall* 48,33% dengan waktu 23,6 detik, sedangkan *AlexNet* menghasilkan nilai *accuracy* 32,58%, *precision* 37,17%, dan *recall* 32,58% dengan waktu 132,8 detik. Pada skema kedua, *LeNet* menghasilkan nilai *accuracy* 92,47%, *precision* 92,88%, dan *recall* 92,47% dengan waktu 230 detik, sedangkan *AlexNet* menghasilkan nilai *accuracy* 91,62%, *precision* 92,07%, dan *recall* 91,60% dengan waktu 1.355,4 detik.

4.2 Saran

Pada penelitian selanjutnya dapat dilakukan dengan menambah jumlah data atau menambah data dari sumber berbeda yang melalui proses segmentasi atau tidak, menggunakan arsitektur lain, mengubah jenis fungsi aktivasi, mengubah jenis pooling layer, dan mengubah jenis *optimizer* untuk mendapatkan hasil yang lebih baik.

Daftar Pustaka

- [1] Shivashankara and Srinath, "American Sign Language Recognition System: An Optimal Approach," *Int. J. Image, Graph. Signal Process.*, vol. 10, no. 8, pp. 18–30, 2018, doi: 10.5815/ijigsp.2018.08.03.
- [2] T. Hunt, J. Carper, T. Lasley, C. Raisch, and E. Drasgow, "American Sign Language," *Encycl. Educ. Reform Dissent*, 2013, doi: 10.4135/9781412957403.n31.
- [3] M. E. Al Rivan, H. Irsyad, K. Kevin, and A. T. Narta, "Pengenalan Alfabet American Sign Language Menggunakan K-Nearest Neighbors Dengan Ekstraksi Fitur Histogram Of Oriented Gradients," *J. Tek. Inform. dan Sist. Inf.*, vol. 5, no. 3, pp. 328–339, 2020, doi: 10.28932/jutisi.v5i3.1936.
- [4] I. Fareza, R. Busdin, M. E. Al Rivan, and H. Irsyad, "Pengenalan Alfabet Bahasa Isyarat Amerika Menggunakan Edge Oriented Histogram dan Image Matching," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, no. 1, pp. 82–92, 2018, doi: 10.28932/jutisi.v4i1.747.
- [5] M. Ezar, A. Rivan, and M. T. Noviardy, "Klasifikasi American Sign Language Menggunakan Ekstraksi Fitur Histogram of Oriented Gradients dan Jaringan Syaraf Tiruan," vol. 6, pp. 442–451, 2020.
- [6] M. Zufar and B. Setiyono, "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time," *J. Sains dan Seni ITS*, vol. 5, no. 2, p. 128862, 2016, doi: 10.12962/j23373520.v5i2.18854.
- [7] M. Bagus, S. Bakti, and Y. M. Pranoto, "Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network," pp. 11–16, 2019.
- [8] M. Swapna, Y. K. Sharma, and B. M. G. Prasad, "CNN Architectures: Alex Net, Le Net, VGG, Google Net, Res Net," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 953–960, 2020, doi: 10.35940/ijrte.f9532.038620.
- [9] S. Kulkarni and S. Harnoorkar, "Comparative Analysis of CNN Architectures," vol. 7, no. June 6, pp. 1459–1464, 2020.
- [10] M. Kayed, A. Anter, and H. Mohamed, "Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture," *Proc. 2020 Int. Conf. Innov. Trends Commun. Comput. Eng. ITCE 2020*, no. February, pp. 238–243, 2020, doi: 10.1109/ITCE48509.2020.9047776.
- [11] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018, doi: 10.32528/JUSTINDO.V3I2.2254.
- [12] V. Maeda-Gutiérrez *et al.*, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041245.
- [13] Akash, "ASL Alphabet," <https://www.kaggle.com/grassknotted/asl-alphabet/>, 2018. .