



Pengenalan rambu lalu lintas menggunakan convolutional neural networks

Traffic sign recognition using convolutional neural networks

Mutaqin Akbar^{*)}

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta
Gg. Jembatan Merah No.84C, Kampus 2 UMBY, Sleman, Indonesia 55283

Cara sitasi: M. Akbar, "Pengenalan rambu lalu lintas menggunakan convolutional neural networks," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 120-125, 2021. doi: [10.14710/jtsiskom.2021.13959](https://doi.org/10.14710/jtsiskom.2021.13959), [Online].

Abstract – Traffic sign recognition (TSR) can be used to recognize traffic signs by utilizing image processing. This paper presents traffic sign recognition in Indonesia using convolutional neural networks (CNN). The overall image dataset used is 2050 images of traffic signs, consisting of 10 kinds of signs. The CNN layer used in this study consists of one convolution layer, one pooling layer using maxpool operation, and one fully connected layer. The training algorithm used is stochastic gradient descent (SGD). At the training stage, using 1750 training images, 48 filters, and a learning rate of 0.005, the recognition results in 0.005 of loss and 100 % of accuracy. At the testing stage using 300 test images, the system recognizes the signs with 0.107 of loss and 97.33 % of accuracy.

Keywords – convolutional neural networks; traffic sign; sign recognition; image processing

Abstrak – Traffic sign recognition (TSR) digunakan mengenali rambu lalu lintas dengan memanfaatkan pengolahan citra. Artikel ini menyajikan pengenalan rambu lalu lintas di Indonesia menggunakan convolutional neural networks (CNN). Dataset citra yang digunakan secara keseluruhan adalah 2050 citra rambu lalu lintas, yang terdiri dari 10 macam rambu. Lapisan CNN yang digunakan terdiri dari satu lapisan konvolusi, satu lapisan pooling menggunakan operasi maxpool, dan satu lapisan fully-connected. Algoritme pelatihan yang digunakan adalah Stochastic Gradient Descent (SGD). Pada tahap pelatihan dengan menggunakan 1750 data citra latih, 48 filter dan laju pelatihan 0,005, dihasilkan galat 0,005 dan akurasi 100 %. Pada tahap pengujian menggunakan 300 data citra uji, sistem dapat mengenali rambu lalu lintas dengan galat 0,107 dan akurasi mencapai 97,33 %.

Kata kunci – convolutional neural networks; rambu lalu lintas; pengenalan rambu; pengolahan citra

I. PENDAHULUAN

Rambu lalu lintas diatur menurut Peraturan Menteri Perhubungan Nomor 13 tahun 2014 dan merupakan bagian dari perlengkapan jalan yang di dalamnya terdapat angka, huruf, kalimat, lambing dan/atau perpaduan di antaranya. Rambu lalu lintas digunakan sebagai pemberi larangan, peringatan, perintah, dan petunjuk bagi semua pemakai jalan. Berdasarkan data dari Dinas Perhubungan, terdapat lebih dari 300 rambu lalu lintas yang ada di Indonesia, terdiri dari berbagai jenis rambu, seperti rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk.

Traffic sign recognition (TSR) atau pengenalan rambu lalu lintas adalah teknologi yang dapat mengenali rambu lalu lintas dengan memanfaatkan pengolahan citra. TSR telah diaplikasikan pada *driver assistance systems* (DAS) [1], *advance driver assistance systems* (ADAS) [2], [3], *autonomous driving systems* (ADS) [4], pemahaman suasana perkotaan, keamanan jalan raya, dan pemantauan rambu untuk kepentingan perawatan. Keluaran dari TSR dapat menjadi masukan, baik untuk DAS, ADAS, ADS, maupun sistem lainnya. Akurasi keluaran TSR akan sangat mempengaruhi keputusan yang diambil oleh sistem-sistem tersebut. Kesalahan pengambilan keputusan akan berdampak fatal dan yang paling buruk adalah terjadinya kecelakaan lalu lintas yang menimbulkan korban jiwa.

Beberapa pendekatan maupun metode sudah menggunakan komputasi matriks yang mahal secara komputasi dalam menyelesaikan masalah deteksi dan pengenalan secara serentak. Di awal, algoritme sederhana dan secara komputasi ringan, seperti *thresholding* warna, digunakan sebagai deteksi. Klasifikasi kemudian dilakukan pada kandidat hasil deteksi dengan algoritme yang lebih mahal secara komputasi dan lebih akurat.

Dalam pengenalan rambu di Indonesia, beberapa pendekatan yang digunakan antara lain adalah penggabungan *support vector machine* (SVM) dan *K-nearest neighbors* (KNN) yang memperoleh akurasi pengenalan 93,9 % [5] dan *template matching* dengan akurasi pengenalan 88 % [6]. Lebih lanjut, klasifikasi dengan jarak City-block digunakan untuk mengenali rambu lalu lintas berbentuk lingkaran yang memperoleh akurasi pengenalan 88 % dalam [7], *shape detection* untuk

^{*)} Penulis korespondensi (Mutaqin Akbar)
Email: mutaqin@mercubuana-yogya.ac.id

mengenali kategori rambu lalu lintas dengan akurasi pengenalan 80,7 % [8], *circle Hough transform* (CHT) dan *convolutional neural network* (CNN) dengan akurasi pengenalan masing-masing 61,3 % dan 75 % [9].

Pendekatan CNN yang telah dilakukan tersebut belum mampu mendapatkan nilai akurasi yang maksimal [9]. Hal ini dapat disebabkan karena terjadi *overfitting* pada model. Artikel ini mengusulkan pendekatan CNN yang masih bisa dimaksimalkan nilai akurasi pengenalannya dengan menggunakan dataset berjumlah banyak untuk menghindari model yang tidak tergeneralisir sehingga dapat menyebabkan *overfitting* [10].

II. METODE PENELITIAN

A. Dataset

Rambu lalu lintas yang digunakan adalah sejumlah 10 macam dan terdiri dari tiga bentuk, yaitu lingkaran, belah ketupat dan oktagon. Rambu tersebut merupakan campuran dari rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk.

Pengambilan data citra dilakukan dalam dua cara, yaitu pengambilan secara langsung dengan menggunakan kamera dan pengambilan secara tidak langsung melalui aplikasi peta. Data citra diambil dari berbagai sudut pandang pada siang dan malam hari. Hasil pengambilan citra berupa citra berwarna, yang kemudian dipotong pada *region of interest* (ROI) dengan ukuran piksel citra 128x128. Jumlah data citra yang digunakan adalah 2050 data citra yang terdiri dari 1750 data citra latih dan 300 data citra uji. Contoh dataset citra dapat dilihat pada Tabel 1. Semua data citra diubah ukuran pikselnya dari 128x128 ke 28x28 dan dikonversi ke dalam ruang warna aras keabuan.




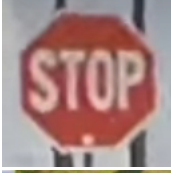



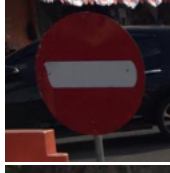
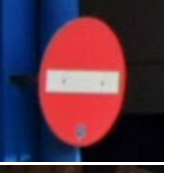




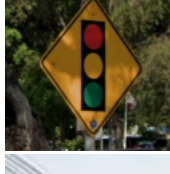






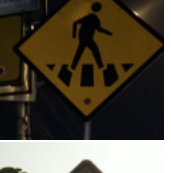





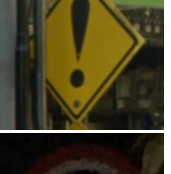


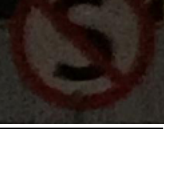
B. Convolutional neural network (CNN)

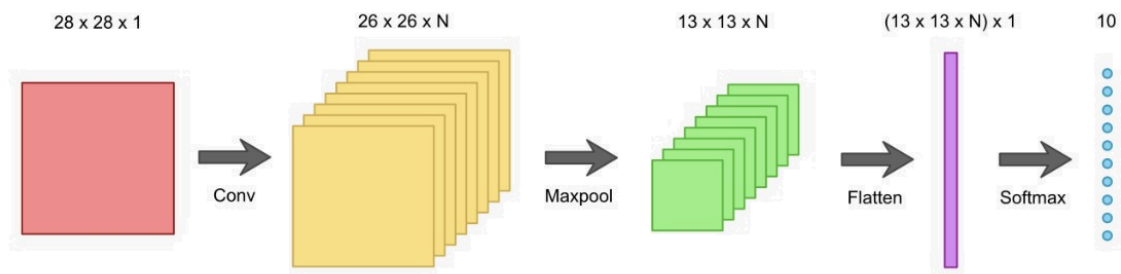
CNN digunakan sebagai algoritme pelatihan untuk memahami muatan sebuah citra dan sudah menunjukkan kinerja yang baik pada segmentasi citra, klasifikasi citra, deteksi citra dan tugas-tugas sejenisnya [11], [12]. CNN terdiri dari 3 lapisan utama (selain lapisan masukan dan keluaran), yaitu lapisan konvolusi, lapisan *pooling*/penggabungan dan lapisan *fully-connected*.

CNN yang digunakan dalam kajian ini termasuk ke dalam *supervised learning neural network*, dimana membutuhkan sekumpulan data latih yang terdiri dari masukan dan target yang terkait dengan setiap masukan (Gambar 1). Jaringan syaraf tiruan target digunakan untuk menentukan seberapa baik pelatihannya dan sebagai panduan penyesuaian pada nilai bobot untuk mengurangi galat secara keseluruhan [13].

Lapisan pertama dari CNN adalah lapisan konvolusi untuk menggabungkan dua sinyal, yaitu sinyal masukan dan sinyal kernel filter, untuk membentuk sinyal ketiga, yaitu sinyal keluaran [14]. Masukan pada lapisan konvolusi merupakan data citra aras keabuan berukuran piksel 28x28. Jumlah filter yang digunakan pada lapisan konvolusi beragam dari 8, 16, 32, dan seterusnya. Filter berukuran 3x3 dengan setiap filter diinisiasi dengan

Tabel 1. Contoh dataset citra rambu lalu lintas

No	Contoh 1	Contoh 2	Contoh 3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			



Gambar 1. Arsitektur CNN yang digunakan (N adalah jumlah filter)

nilai acak. Filter berjalan melalui piksel pada citra mulai koordinat 2, 2 sampai 27, 27 (koordinat dimulai dari 1) dan menghasilkan keluaran berukuran piksel 26x26, sehingga keluaran pada lapisan ini adalah sebuah matriks berukuran 26x26xN filter.

Matriks keluaran dari lapisan konvolusi masuk ke dalam lapisan penggabungan untuk mencapai invariansi spasial dengan mengurangi resolusi matriks keluaran dari lapisan konvolusi. Lapisan penggabungan mengurangi resolusi matriks dengan meringkas neuron dari ketetanggaan spasial yang kecil [15]. Operasi yang digunakan pada lapisan ini adalah penggabungan dengan mempertimbangkan nilai maksimal, dimana diterapkan fungsi jendela 2x2 untuk menghitung nilai maksimal dari ketetanggaan piksel seukuran fungsi jendela tersebut. Keluaran dari lapisan ini adalah matriks berukuran setengah dari ukuran matriks masukannya, yaitu 13x13xN filter.

Matriks keluaran dari lapisan penggabungan diubah menjadi sebuah vektor sebelum masuk ke dalam lapisan *fully-connected* atau proses perataan. Proses ini mengubah matriks berukuran 13x13xN filter menjadi vektor berukuran (13x13xN filter)x1. Vektor hasil proses perataan diteruskan ke lapisan *fully-connected* dimana akan dilakukan proses klasifikasi dengan keluaran berjumlah 10 kelas (sesuai dengan macam rambu lalu lintas). Fungsi aktivasi softmax digunakan untuk memprediksi distribusi probabilitas dari kelas-kelas yang ada [16]. Setelah melalui proses prediksi galat dari prediksi tersebut dicari berdasarkan target yang akan dituju menggunakan fungsi galat *cross-entropy*. Semakin kecil galat, maka semakin baik pengenalannya. Langkah-langkah dalam fase tersebut di atas dilakukan secara runut maju.

C. Backward propagation

Pelatihan CNN dilakukan dengan melakukan pengaturan terhadap parameter untuk meminimalisir fungsi galat. Pengaturan dan penyesuaian bobot bergantung pada pengaturan bobot dengan fungsi galat terendah (penurunan gradien) dan arah perubahan kemiringan selama penurunan gradien sesuai [17].

Selama fase runut maju, setiap lapisan akan menyimpan data, seperti masukan, dan nilai antara, yang diperlukan untuk fase runut mundur. Setiap fase runut mundur didahului oleh fase runut maju yang sesuai. Selama fase runut mundur, setiap lapisan

menerima gradien galat terhadap keluarannya dan juga mengembalikan gradien galat terhadap masukannya.

Pada lapisan *fully-connected*, masukannya adalah gradien galat dari keluaran fungsi aktivasi softmax. Data yang dibutuhkan adalah data yang tersimpan pada fase runut maju, yaitu vektor masukan dan hasil perkalian titik antara vektor masukan dan bobot ditambah dengan bias atau penjumlahan terbobot. Setelah itu, perhitungan gradien dari nilai keluaran dilakukan terhadap penjumlahan terbobot. Perhitungan gradien dari penjumlahan terbobot dilakukan terhadap bobot, bias, dan masukan, gradien dari galat dilakukan terhadap penjumlahan terbobot dan gradien dari galat dilakukan terhadap bobot, bias, dan masukan. Bobot dan bias kemudian diperbarui dengan mempertimbangkan laju pelatihan. Gradien dari galat terhadap masukan akan menjadi keluaran dari lapisan ini dan masukan untuk lapisan penggabungan.

Lapisan penggabungan pada fase runut mundur tidak melakukan pelatihan karena di dalam lapisan ini tidak terdapat bobot. Pada fase runut maju, ukuran keluaran adalah setengah dari masukan, sehingga pada fase runut mundur dilakukan penggandaan lebar dan tinggi gradien galat dengan menempatkan setiap nilai gradien ke tempat nilai maksimal asli pada blok 2x2 yang sesuai dan nilai lain akan diatur menjadi 0.

Lapisan konvolusi pada fase runut mundur, masukannya adalah gradien yang ditempatkan pada nilai maksimal. Data tersimpan pada lapisan ini yang dibutuhkan pada fase runut mundur adalah masukan pada fase runut maju. Perhitungan gradien galat dilakukan terhadap filter. Filter yang digunakan diperbarui dengan mempertimbangkan perhitungan gradien dan laju pelatihan.

III. HASIL DAN PEMBAHASAN

A. Pelatihan

Pada tahap pelatihan, parameter yang digunakan adalah jumlah filter 16, 32 dan 48, laju pelatihan 0,005 dan 0,0075, dan jumlah *epoch* atau iterasi 40. Pengujian dilakukan untuk setiap model yang dihasilkan dari pelatihan tersebut. Hasil pelatihan dinyatakan dalam Tabel 2. Pelatihan dengan laju pelatihan 0,005 dan jumlah filter 16 memperoleh galat yang stabil turun hingga 0.006 pada iterasi ke-40. Akurasi mengalami fluktuasi. Di awal iterasi adalah 81,77 % dan naik

Tabel 2. Hasil pelatihan untuk setiap parameter yang digunakan (akurasi dalam %)

epoch	laju pelatihan=0.005						laju pelatihan=0.0075					
	filter=16		filter=32		filter=48		filter=16		filter=32		filter=48	
	galat	akurasi	galat	akurasi	galat	akurasi	galat	akurasi	galat	akurasi	galat	akurasi
1	0,809	81,77	0,713	83,89	0,681	84,57	0,605	86,86	0,564	87,71	0,56	87,89
2	0,418	89,37	0,406	89,6	0,403	89,89	0,325	91,71	0,314	92,06	0,318	91,66
4	0,167	95,66	0,164	95,77	0,16	96,06	0,137	96,29	0,134	96,51	0,132	96,69
6	0,103	98,06	0,101	97,6	0,097	97,94	0,084	97,94	0,083	98,06	0,079	98,11
8	0,069	98,69	0,066	98,74	0,063	99,09	0,051	99,37	0,053	99,14	0,048	99,37
10	0,046	99,66	0,044	99,49	0,042	99,66	0,032	99,71	0,032	99,71	0,03	99,77
12	0,033	99,71	0,032	99,71	0,03	99,77	0,026	99,77	0,022	99,89	0,021	99,94
14	0,025	99,83	0,024	99,89	0,022	99,94	0,017	100	0,017	99,89	0,017	99,94
16	0,019	100	0,018	99,94	0,018	100	0,016	99,89	0,015	99,83	0,014	99,94
18	0,015	100	0,015	100	0,015	100	0,013	99,89	0,011	100	0,012	99,89
20	0,013	100	0,013	100	0,013	100	0,013	99,77	0,01	99,89	0,011	99,94
22	0,012	100	0,011	100	0,011	100	0,011	99,89	0,013	99,83	0,008	99,94
24	0,01	100	0,01	100	0,01	100	0,008	99,94	0,008	100	0,026	99,6
26	0,009	100	0,009	100	0,009	100	0,007	100	0,007	99,94	0,008	99,94
28	0,009	100	0,009	100	0,008	100	0,007	99,94	0,009	99,83	0,007	99,89
30	0,008	100	0,008	100	0,008	100	0,007	99,94	0,013	99,6	0,006	99,94
32	0,007	99,94	0,007	100	0,007	100	0,013	99,66	0,007	99,94	0,005	99,94
34	0,007	100	0,007	100	0,006	100	0,006	99,89	0,014	99,6	0,005	100
36	0,006	100	0,007	100	0,006	100	0,006	99,94	0,006	99,83	0,014	99,89
38	0,006	100	0,006	100	0,006	100	0,004	100	0,007	99,89	0,006	99,89
40	0,006	99,94	0,006	100	0,005	100	0,004	100	0,005	100	0,004	100

mencapai 100% pada iterasi ke-16, kemudian turun menjadi 99,94 % pada iterasi ke-32 dan 40. Dengan parameter jumlah filter 32, masih didapatkan galat yang stabil turun dan tidak terjadi fluktuasi dengan nilai di iterasi ke-40 yaitu 0,006. Akurasi juga tidak mengalami fluktuasi dibanding dengan jumlah filter 16, dan stabil naik hingga 100 % mulai iterasi ke-18 sampai iterasi terakhir. Pada jumlah filter 48, galat terlihat stabil turun dan mencapai galat 0,005 pada iterasi ke-40. Akurasi juga terlihat stabil naik dan mencapai 100 % pada iterasi ke-16 sampai iterasi terakhir.

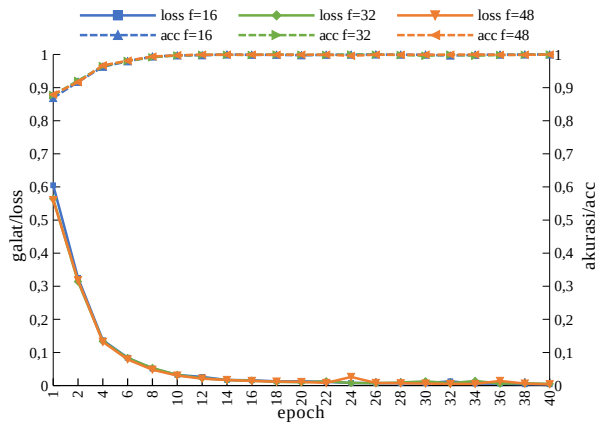
Dengan laju pelatihan 0,0075 dan jumlah filter 16, fluktuasi galat terjadi pada iterasi ke-32 dan kembali turun hingga mencapai 0,004 pada iterasi ke-40. Akurasi juga terlihat terjadi fluktuasi dan sempat mencapai nilai 100 % pada iterasi ke-14, 26, 38, dan 40. Pada jumlah filter 32, fluktuasi nilai galat terjadi pada iterasi ke-22, 30, dan 34, yang akhirnya mencapai nilai 0,005 pada epoch terakhir. Akurasi juga terlihat terjadi fluktuasi dan sempat mencapai nilai 100 % pada iterasi ke-18, 24, dan 40. Pada jumlah filter 48, masih terlihat ada fluktuasi galat pada iterasi ke-24 dan 36, yang akhirnya mencapai nilai 0,004 pada iterasi terakhir. Akurasi juga masih terlihat terjadi fluktuasi dan sempat mencapai nilai 100 % pada iterasi ke-34 dan 40.

Grafik galat dan akurasi proses pelatihan dengan laju 0,005 ditunjukkan pada Gambar 2. Fluktuasi akurasi terjadi pada jumlah filter 16 dan tidak terjadi pada jumlah filter 32 dan 48. Galat terlihat stabil turun, menandakan laju pelatihan yang digunakan sudah baik dan mampu meraih konvergensi. Grafik galat dan nilai akurasi dengan laju 0,0075 ditunjukkan pada Gambar 3. Fluktuasi galat dan akurasi terjadi pada semua jumlah filter yang dipakai. Hal ini menunjukkan laju pelatihan yang digunakan kurang baik, walaupun pada akhir iterasi mampu meraih akurasi pelatihan 100 % untuk semua filter yang digunakan.

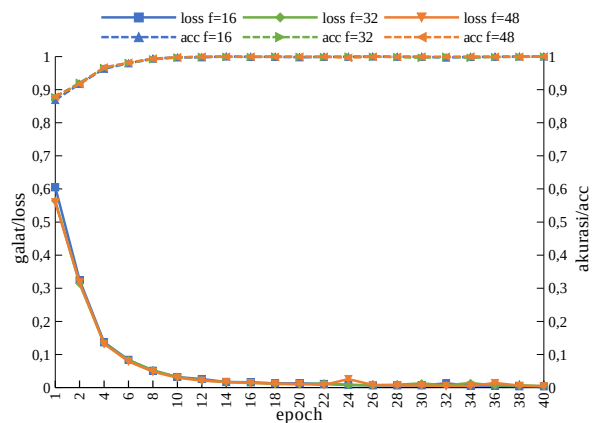
B. Pengujian

Hasil pengujian untuk setiap parameter dinyatakan dalam Tabel 3. Model dengan laju pelatihan 0,005 dan jumlah filter 16 menghasilkan galat 0,108 dan akurasi 97,33 %. Jumlah filter 32, meskipun pada proses pelatihan lebih baik dari jumlah filter 16, menghasilkan galat lebih besar yaitu 0,134 dan nilai akurasi lebih kecil yaitu 96,33 %. Pada jumlah filter 48, galat lebih baik dari filter 16 dan 32, yaitu 0,107 dan nilai akurasi 97,33 % atau sama dengan jumlah filter 16.

Walaupun melalui proses pelatihan yang fluktuatif baik nilai galat maupun nilai akurasi, model dengan laju



Gambar 2. Grafik galat dan akurasi dengan laju pelatihan 0,005



Gambar 3. Grafik galat dan akurasi dengan laju pelatihan 0,0075

pelatihan 0.0075 dan jumlah filter 16 mampu mendapatkan galat 0,129 dan nilai akurasi 97 %. Pada jumlah filter 32 didapatkan galat 0,116 dan nilai akurasi 97,33 %. Dengan jumlah filter 48, nilai galat dan nilai akurasi terburuk diperoleh dari semua parameter yang ada, yaitu galat 0,162 dan akurasi sebesar 94,67 %.

Berdasarkan Tabel 4, akurasi TSR menggunakan CNN yang diusulkan dalam kajian ini, mempunyai akurasi yang lebih baik daripada metode lain pada rambu lalu lintas di Indonesia untuk semua laju pelatihan. Metode CNN yang diterapkan ini menghasilkan akurasi terbaik 97,33 % lebih tinggi dari metode CNN dalam [9] dengan akurasi 75 %. Namun, dalam [9], sistem diterapkan pada subjek yang berjalan, sedangkan dalam kajian ini dilakukan di objek rambu yang diam. Metode CNN ini menghasilkan akurasi lebih baik dalam pengenalan rambu lalu lintas daripada SVM dan KNN dalam [5], Template matching dalam [6], klasifikasi dengan jarak City-block dalam [7], Shape detection dalam [8], dan CHT dalam [9].

IV. KESIMPULAN

CNN telah berhasil mengenali rambu lalu lintas di Indonesia yang dilakukan terhadap 1750 data citra latih dan 300 data citra uji. Parameter pelatihan terbaik dengan akurasi di akhir iterasi mencapai 100 %, tidak terjadi fluktuasi nilai galat dan akurasi, dan nilai akurasi pengujian terbaik adalah laju pelatihan 0,005 dan jumlah filter 48 dengan nilai galat 0,107 dan akurasi pengenalan mencapai 97,33 %.

DAFTAR PUSTAKA

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014. doi: [10.1109/MITS.2014.2336271](https://doi.org/10.1109/MITS.2014.2336271)

[2] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced driver-assistance systems: a path toward autonomous vehicles," *IEEE*

Tabel 3. Hasil pengujian untuk parameter laju pelatihan dan filter

Laju pelatihan	Galat			Akurasi (%)		
	f=16	f=32	f=48	f=16	f=32	f=48
0,005	0,108	0,134	0,107	97,33	96,33	97,33
0,0075	0,129	0,116	0,162	97	97,33	94,67

Tabel 4. Perbandingan akurasi metode TSR di Indonesia

Referensi	Metode	Akurasi (%)
[5]	SVM dan KNN	93,9
[6]	Template matching	88
[7]	Klasifikasi City-block	88
[8]	Shape detection	80,7
[9]	CHT	61,3
[9]	CNN	75
Kajian ini	CNN (diusulkan)	97,33

Consumer Electronics Magazine, vol. 7, no. 5, pp. 18–25, 2018. doi: [10.1109/MCE.2018.2828440](https://doi.org/10.1109/MCE.2018.2828440)

[3] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of advanced driver assistance systems (ADAS)," in *AIP Conference Proceedings*, vol. 1906, 120002, 2017. doi: [10.1063/1.5012394](https://doi.org/10.1063/1.5012394)

[4] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 163–168. doi: [10.1109/IVS.2011.5940562](https://doi.org/10.1109/IVS.2011.5940562)

[5] C. Rahmad, I. Fauziah Rahmah, and R. Andrie Asmara, "Deteksi dan pengenalan rambu lalu lintas di Indonesia menggunakan RGBN dan Gabor," in *Prosiding SENTRINOV*, vol. 3, no. 1, pp. T113-22, 2017.

[6] T. O. Chrisdwianto, H. Fitriyah, and E. Rosana Widasari, "Perancangan sistem deteksi dan pengenalan rambu peringatan menggunakan metode template matching," *Jurnal*

- Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 3, pp. 1265–1274, 2018.
- [7] G. Romadhon and M. Murinto, “Aplikasi pengenalan citra rambu lalu lintas berbentuk lingkaran menggunakan metode jarak city-block,” *Jurnal Sarjana Teknik Informatika*, vol. 2, no. 2, pp. 286–294, 2014.
- [8] O. R. Sitanggang, H. Fitriyah, and F. Utaminingrum, “Sistem deteksi dan pengenalan jenis rambu lalu lintas menggunakan metode shape detection pada Raspberry Pi,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 6108–6117, 2018.
- [9] A. Triyadi and F. Utaminingrum, “Pengembangan sistem rekognisi rambu kecepatan menggunakan circle hough transform dan convolutional neural network berbasis Raspberry Pi,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 56–64, 2020.
- [10] X. Ying, “An Overview of overfitting and its solutions,” *Journal of Physics: Conference Series*, vol. 1168, 022022, 2019. doi: [10.1088/1742-6596/1168/2/022022](https://doi.org/10.1088/1742-6596/1168/2/022022)
- [11] Y. Le Cun *et al.*, “Handwritten digit recognition: applications of neural net chips and automatic learning,” in *Neurocomputing*, F. F. Soulié and J. Héroult, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 303–318. doi: [10.1007/978-3-642-76153-9_35](https://doi.org/10.1007/978-3-642-76153-9_35)
- [12] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artificial Intelligent Review*, vol. 53, no. 8, pp. 5455–5516, 2020. doi: [10.1007/s10462-020-09825-6](https://doi.org/10.1007/s10462-020-09825-6)
- [13] A. P. Engelbrecht, *Computational intelligence: an introduction*, 2nd ed. Chichester, England; Hoboken, NJ: John Wiley & Sons, 2007.
- [14] S. W. Smith, *The scientist and engineer’s guide to digital signal processing*. California: California Technical Publishing, 1999.
- [15] D. Hutchison *et al.*, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Artificial Neural Networks – ICANN 2010*, vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101. doi: [10.1007/978-3-642-15825-4_10](https://doi.org/10.1007/978-3-642-15825-4_10)
- [16] P. Sadowski, “Notes on backpropagation.” 2016, Accessed: Oct. 15, 2020. [Online]. Available: <https://www.ics.uci.edu/~pjsadows/notes.pdf>
- [17] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, “A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks,” in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Belgaum, India, Dec. 2018, pp. 92–99, doi: [10.1109/CTEMS.2018.8769211](https://doi.org/10.1109/CTEMS.2018.8769211)



©2021. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).