

Multi-Scale Convolutional Neural Networks untuk Pengenalan Rambu Lalu Lintas di Indonesia

Mutaqin Akbar^{[1]*}, Agus Sidiq Purnomo^[2], Supatman^[3]

Program Studi Informatika^{[1],[2],[3]}
Universitas Mercu Buana Yogyakarta
Sleman, Indonesia

mutaqin@mercubuana-yogya.ac.id^[1], sidiq@mercubuana-yogya.ac.id^[1], supatman@mercubuana-yogya.ac.id^[1]

Abstract— Traffic sign recognition, which is often abbreviated to as TSR, is used to recognize traffic signs. TSR itself can be applied to driver assistance systems (DAS), advanced driver assistance systems (ADAS), autonomous driving systems (ADS), urban environment, road safety, and sign monitoring for maintenance purposes. TSR in Indonesia has been researched many times before, where the best accuracy result obtained is 99,33%. This article presents an update of the traffic sign recognition in Indonesia using multi-scale convolutional neural network (CNN). The dataset used in this research is taken using several ways namely live image capture, indirect image capture using Google Maps, image capture from various angles and image capture in day and night. The dataset collected amounted to 2050 traffic sign images which are grouped into 10 classes. Dataset is then cropped based on region of interest (ROI), resized to 28x28 and converted into grayscale colorspace. The CNN model consists of three 3x3 convolution layers, three 2x2 convolution layers (Maxpool) and one fully-connected layer using softmax activation function. The number of filters used in each convolution layer is 32. The training algorithm used is Stochastic gradient descent (SGD). Using 1750 training image data, 20 epochs and a learning rate of 0,005, the loss and accuracy value obtained at the training stage respectively 0,0026 and 100%. And at the testing stage using 300 test data, the loss value obtained is 0.017 and the accuracy value obtained is 99,67%.

Keywords— multi-scale convolutional neural networks, traffic sign, traffic sign recognition

Abstrak— Teknologi pengenalan rambu lalu lintas yang sering disebut dengan *traffic sign recognition* (TSR), digunakan untuk mengenali rambu lalu lintas melalui pemanfaatan pengolahan citra. TSR sendiri dapat diaplikasikan pada sistem pembantu pengemudi, sistem pembantu pengemudi tingkat lanjut, sistem mengemudi otonom, keamanan jalan raya, pemahaman suasana perkotaan, dan pemantauan rambu untuk kepentingan perawatan. Pengenalan rambu lalu lintas di Indonesia telah dikaji beberapa kali oleh beberapa peneliti sebelumnya, dimana hasil akurasi terbaik yang pernah dicapai yaitu 99,33%. Perbaruan dari pengenalan rambu lalu lintas di Indonesia menggunakan *multi-scale convolutional neural network* (CNN) telah disajikan pada artikel ini. Dataset yang digunakan pada penelitian ini diambil dengan beberapa cara diantaranya pengambilan langsung, pengambilan tidak langsung dengan *Google Maps*, pengambilan dari berbagai sudut dan pengambilan

pada siang dan malam hari. Dataset yang dikumpulkan berjumlah 2050 data citra rambu lalu lintas yang dikelompokkan kedalam 10 kelas. Dataset dipotong pada *region of interest* (ROI), diubah ukurannya menjadi 28x28 dan dikonversi ke ruang warna aras keabuan. Model CNN terdiri dari tiga lapisan konvolusi berukuran 3x3, tiga lapisan penggabungan (*Maxpool*) berukuran 2x2 dan satu lapisan *fully-connected* yang memanfaatkan fungsi aktivasi *Softmax*. Jumlah filter yang digunakan pada setiap lapisan konvolusi adalah 32. Algoritma pelatihan yang digunakan yaitu *Stochastic gradient descent* (SGD). Dengan menggunakan 1750 data citra latih, nilai epoch 20, dan laju pelatihan 0,005, nilai galat dan nilai akurasi yang didapatkan pada tahap pelatihan adalah masing-masing 0,0026 dan 100%. Sedangkan pada tahap pengujian, dengan 300 data citra uji, model CNN mampu memperoleh nilai galat 0,017 dan nilai akurasi mencapai 99,67%.

Kata Kunci— multi-scale convolutional neural networks, pengenalan rambu lalu lintas, rambu lalu lintas

I. PENDAHULUAN

Rambu lalu lintas merupakan unsur kelengkapan jalan yang didalamnya terdapat angka, huruf, kalimat, lambang dan/atau kombinasi diantaranya yang dapat berfungsi sebagai larangan, peringatan, perintah, atau petunjuk bagi pengguna jalan. Di Indonesia sendiri, terdapat ratusan rambu lalu lintas yang terdiri berbagai jenis rambu lalu lintas sesuai fungsi dan kebutuhannya, yaitu rambu anjuran, rambu larangan, rambu peringatan, rambu perintah dan rambu petunjuk. Kementerian Perhubungan mengatur tentang rambu lalu lintas melalui Peraturan Menteri Perhubungan tentang Rambu Lalu Lintas Nomor 13 Tahun 2014[1].

Teknologi pengenalan rambu lalu lintas yang sering disebut TSR, digunakan untuk mengidentifikasi rambu lalu lintas melalui penggunaan pengolahan citra. TSR sendiri telah terapkan pada sistem pembantu pengemudi (DAS)[2], [3], sistem pembantu pengemudi tingkat lanjut (ADAS)[4]–[6], sistem mengemudi otonom (ADS)[7], [8], yang disebut terakhir saat ini sudah mulai dilirik oleh banyak konsumen contohnya mobil listrik yang juga mobil pintar Tesla. TSR juga dapat diimplementasikan untuk keamanan jalan raya, pemahaman suasana perkotaan, dan peninjauan rambu lalu lintas untuk kebutuhan perawatan.

TABEL I CONTOH DATASET

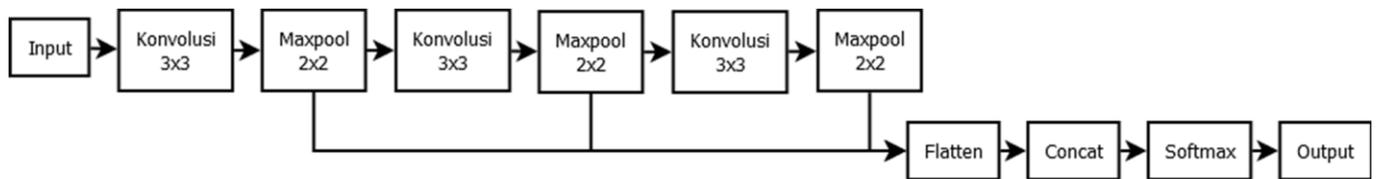
No	Dataset			No	Dataset		
	Contoh 1	Contoh 2	Contoh 3		Contoh 1	Contoh 2	Contoh 3
1				6			
2				7			
3				8			
4				9			
5				10			

Akurasi pengenalan pada TSR sangat mempengaruhi sistem-sistem seperti DAS, ADAS, ADS, maupun sistem lainnya karena sedikit kesalahan pengenalan dan pengambilan keputusan oleh sistem tersebut akan memberikan dampak fatal dan yang terburuk yaitu terjadinya kecelakaan lalu lintas yang menimbulkan korban jiwa dan material.

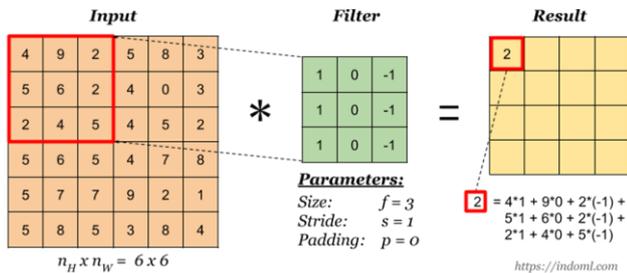
Identifikasi atau deteksi rambu lalu lintas di Indonesia sudah cukup lama dikaji oleh beberapa peneliti. Pada tahun 2014, pengenalan rambu lalu lintas menggunakan metode jarak city-block dilakukan dan mendapatkan akurasi 88%[9]. Kemudian deteksi jenis rambu lalu lintas menggunakan deteksi bentuk juga telah dilakukan dan memperoleh akurasi sebesar 80,7%[10]. Selanjutnya pengenalan rambu lalu lintas dengan memanfaatkan template matching mampu memperoleh akurasi sebesar 88%[11] dan 90,81%[12]. Metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) juga dikaji dan mampu memperoleh akurasi sebesar 93,9%[13]. Algoritma dengan komputasi yang mahal kemudian mulai dikaji, seperti *Circle Hough Transform* (CHT)[14] dan *convolutional neural network* (CNN)[14]-[16]. Dimana CNN

yang diteliti oleh [16] mampu memperoleh tingkat akurasi terbaik dibandingkan dengan algoritma atau metode atau model sebelumnya yaitu 99,33%. Dengan tingkat akurasi yang lebih baik maka teknologi mengemudi otomatis dapat menjadi lebih baik sehingga dapat mengurangi dampak fatal yaitu kecelakaan lalu lintas yang mengakibatkan korban jiwa dan material.

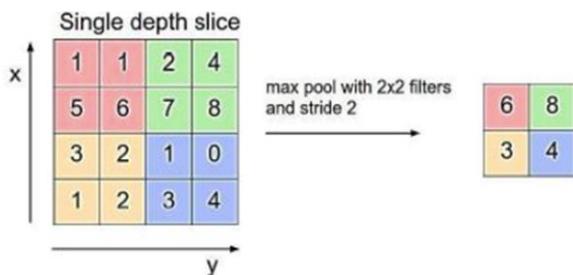
Mengenai algoritma yang disebutkan terakhir, CNN pertama kali dicetuskan oleh Yann LeCun[17], dan perkembangannya sangat pesat hingga saat ini. Salah satu pendekatan pengembangan CNN yang cukup menarik dicetuskan oleh Sermanet dan LeCun yang juga meneliti tentang TSR[18]. Dalam penelitiannya, model CNN dikembangkan dengan tidak hanya keluaran dari lapisan konvolusi terakhir yang menjadi masukan proses klasifikasi, tetapi keluaran setiap lapisan konvolusi dijadikan masukan pada proses klasifikasi. Hal ini menyebabkan fitur-fitur tingkat atas dan juga fitur-fitur tingkat bawah digunakan dalam proses klasifikasi. Proses tersebut diberi nama *multi-scale* oleh Sermanet dan LeCun.



Gambar 3 Arsitektur CNN diusulkan



Gambar 1 Proses konvolusi



Gambar 2 Proses penggabungan

Artikel ini mengusulkan pengembangan terhadap arsitektur CNN yang digunakan untuk pengenalan rambu lalu lintas di Indonesia dengan menggunakan arsitektur CNN Sermanet dan Yann LeCun[18] dan menambahkan lapisan konvolusi dan penggabungan, dimana harapannya arsitektur CNN yang diusulkan mampu meningkatkan akurasi pengenalan rambu lalu lintas yang sudah dikaji sebelumnya.

II. METODOLOGI

A. Dataset

Dalam penelitian yang diusulkan, menggunakan dataset yang terdiri dari 10 macam rambu lalu lintas yang kemudian akan dijadikan target kelas, dimana meliputi rambu lalu lintas dengan bentuk belah ketupat, lingkaran dan oktagon. Rambu lalu lintas ini juga merupakan campuran dari berbagai rambu seperti rambu anjuran, rambu larangan, rambu peringatan, rambu perintah dan rambu petunjuk.

Proses pengambilan dataset dilakukan dengan beberapa cara yaitu yang pertama pengambilan secara langsung dengan bantuan kamera ponsel, kedua yaitu pengambilan secara tidak langsung dengan memanfaatkan hasil tangkapan layar pada aplikasi peta digital (*Google Maps*), ketiga yaitu pengambilan dengan beberapa sudut pandang yang berbeda-beda, keempat yaitu pengambilan yang dilakukan pada siang hari dan pengambilan yang dilakukan pada malam hari.

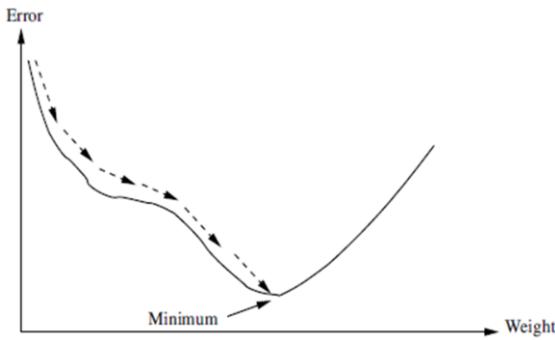
Citra berwarna hasil dari proses pengambilan dataset, kemudian dilakukan proses pemotongan pada daerah yang menjadi perhatian (*region of interest/ROI*) dengan ukuran piksel citra hasil proses pemotongan yaitu 128x128. Tabel I menunjukkan beberapa contoh dari dataset citra rambu lalu lintas yang berhasil dikumpulkan. Jumlah dataset yang berhasil dikumpulkan sejumlah 2050 citra, dengan pembagian dataset meliputi dataset citra latih dan dataset citra uji masing-masing yaitu 1750 citra dan 300 citra.

Sebelum menjadi masukan pada CNN, dataset citra akan diperkecil ukurannya menjadi 28x28 dan juga dilakukan proses perubahan ruang warna dari ruang warna *Red Green Blue* (RGB) ke ruang warna aras keabuan (*grayscale*).

B. Arsitektur CNN

Pada tahun 1989, Yann LeCun dkk[17] memperkenalkan salah satu algoritma pelatihan yang digunakan untuk memahami muatan-muatan pada sebuah citra dan terbukti menunjukkan performa yang baik pada deteksi citra, klasifikasi citra, segmentasi citra dan tugas serupa[19] yang diberi nama CNN. CNN merupakan jaringan saraf tiruan dengan menerapkan pelatihan disupervisi. CNN dilatih menggunakan kumpulan data (sering disebut dataset) yang terdiri dari masukan dan target dimana target tersebut terafiliasi dengan masukan. Target digunakan oleh CNN untuk menentukan seberapa baik performa pelatihan dan sebagai acuan adaptasi nilai bobot untuk mengurangi nilai galat yang diperoleh[20]. CNN terdiri dari beberapa lapisan diantaranya lapisan masukan (*input*), lapisan konvolusi, lapisan pooling (*maxpool*), lapisan *fully-connected* (*flatten, concat, softmax*) dan lapisan keluaran (*output*), seperti Nampak pada Gambar 1.

Lapisan awal pada CNN bernama lapisan konvolusi. Istilah konvolusi sendiri merupakan bentuk matematis dalam memadukan dua sinyal sehingga terbentuk sinyal ketiga[21]. Sinyal pertama biasa disebut sinyal masukan, sedangkan sinyal kedua dalam ranah CNN sering disebut sebagai filter. Dan sinyal terakhir yaitu sinyal ketiga merupakan sinyal keluaran. Sinyal masukan pada lapisan konvolusi untuk pengolahan citra biasanya merupakan citra dengan aras keabuan, walau tidak menutup kemungkinan untuk citra berwarna. Jumlah filter yang digunakan merupakan parameter yang nilainya tidak dibatasi, tetapi biasanya 8, 16, 32 dan seterusnya. Ukuran filter yang digunakan dapat bervariasi dan biasanya berukuran 3x3, 5x5, 7x7, 9x9 dan seterusnya. Ilustrasi proses konvolusi dapat dilihat pada Gambar 2. Pada arsitektur CNN yang diusulkan seperti pada Gambar 1, akan diterapkan 3 lapisan konvolusi dengan berukuran 3x3.



Gambar 5 Ilustrasi penurunan gradien

Selanjutnya, keluaran dari lapisan konvolusi akan melalui lapisan berikutnya yaitu lapisan penggabungan yang dapat dilihat ilustrasinya pada Gambar 3. Lapisan ini bertujuan untuk mencapai invariansi spasial. Hal tersebut dicapai dengan mengurangi resolusi matriks dengan cara memangkas elemen dari ketetanggaan spasial yang memiliki nilai kecil[22]. Tetapi dimungkinkan juga untuk meringkas elemen ketetanggaan dengan cara mencari nilai rerata. Operasi penggabungan yang digunakan pada lapisan penggabungan ini merupakan operasi penggabungan dengan mengambil nilai maksimal dan menyingkirkan nilai yang lebih kecil (*maxpool*). Fungsi jendela yang digunakan untuk mengambil nilai maksimal ketetanggaan spasial yaitu fungsi jendela 2x2. Hal ini menyebabkan keluaran dari lapisan penggabungan merupakan matriks berukuran setengah dari ukuran masukan-nya. Seperti terlihat pada Gambar 1, lapisan penggabungan berada setelah lapisan konvolusi, sehingga terdapat 3 lapisan penggabungan pada arsitektur CNN yang diusulkan.

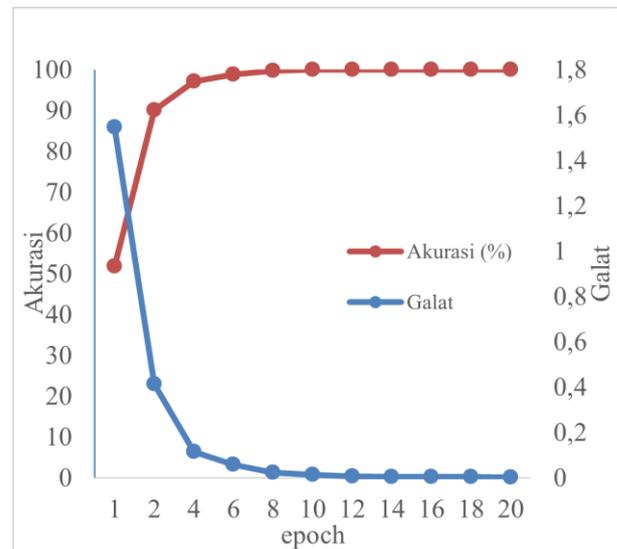
Keluaran dari lapisan penggabungan, selanjutnya akan menjadi masukan pada lapisan *fully-connected*. Tetapi sebelum itu, terdapat proses perataan (*flatten*), dimana keluaran lapisan penggabungan akan diubah menjadi vektor yang akan menjadi masukan pada lapisan *fully-connected*. Keluaran setiap lapisan penggabungan akan melalui proses perataan, sehingga bukan hanya keluaran lapisan penggabungan terakhir saja yang menjadi masukan pada lapisan *fully-connected*, seperti terlihat pada Gambar 1.

Pada lapisan *fully-connected*, keluaran dari proses perataan akan dilakukan proses klasifikasi. Pada proses klasifikasi diperlukan fungsi aktivasi, dimana dapat dipilih dari berbagai opsi seperti linear, step, ramp, sigmoid, hyperbolic, Gaussian, softmax, dan lainnya[20]. Fungsi aktivasi yang digunakan pada lapisan ini yaitu fungsi aktivasi *softmax*. Fungsi tersebut menghitung distribusi probabilitas dari kelas-kelas yang ada[23]. Setelah mendapatkan hasil probabilitas, akan dihitung galat dari perhitungan tersebut berdasarkan pada target yang akan dituju. Hal tersebut dicapai dengan menerapkan fungsi galat *cross-entropy*. Galat yang semakin kecil menunjukkan bahwa prediksi yang dihasilkan semakin meyakinkan.

CNN dilatih dengan melakukan pengaturan dan penyesuaian terhadap parameter yang ada sehingga dicapai nilai galat seminimal mungkin. Penurunan gradien dan arah perubahan kemiringan selama penurunan gradien digunakan untuk mengatur dan menyesuaikan nilai bobot.

TABEL II HASIL PELATIHAN

epoch	Pelatihan	
	Galat	Akurasi (%)
1	1,545	51,77
2	0,412	90,06
3	0,1965	95,54
4	0,1152	97,09
5	0,0747	98,17
6	0,0591	98,8
7	0,0433	98,8
8	0,0239	99,77
9	0,0164	99,77
10	0,0133	99,94
11	0,0085	100
12	0,0071	100
13	0,0062	100
14	0,005	100
15	0,0045	100
16	0,0038	100
17	0,0034	100
18	0,0031	100
19	0,0029	100
20	0,0026	100



Gambar 4 Grafik galat dan akurasi pelatihan

Ilustrasi penurunan gradien dapat dilihat pada Gambar 4. *Stochastic gradient descent* (SGD) akan digunakan untuk melatih CNN, dimana kelebihan dari SGD dibanding penurunan gradien total yaitu konvergen lebih cepat dan mampu untuk keluar dari *local minimum*[24].

III. HASIL DAN PEMBAHASAN

Bagian ini akan dijabarkan dalam dua subbagian yaitu subbagian pertama pelatihan dan subbagian kedua pengujian. Bahasa pemrograman yang digunakan adalah *python* dengan memanfaatkan pustaka *tensorflow*. Pelatihan model CNN dilakukan menggunakan *Graphical Processing Unit* (GPU) untuk mempersingkat waktu pelatihan. GPU yang digunakan yaitu NVIDIA GeForce GTX750Ti.

A. Pelatihan

Parameter-parameter yang digunakan dalam pelatihan model CNN terdiri dari laju pelatihan 0,005 dan jumlah *epoch* 20. Sedangkan beberapa hal yang akan dilatih yaitu filter konvolusi dan bias-nya serta bobot dan bias lapisan *fully-connected*.

Seperti terlihat pada Gambar 1 bahwa arsitektur CNN yang digunakan tersusun atas 3 lapisan konvolusi yang diikuti dengan lapisan penggabungan *Maxpool*. Jumlah filter yang digunakan pada setiap layar konvolusi adalah 32 buah dimana kesemua filter tersebut berukuran 3x3. Sedangkan pada lapisan penggabungan *Maxpool*, digunakan fungsi jendela 2x2.

Setiap keluaran dari lapisan penggabungan *Maxpool* akan melalui proses perataan. Setelah melewati proses perataan, vektor-vektor yang dihasilkan akan digabungkan menjadi sebuah vektor yang terjadi pada lapisan *Concat*. Setelah itu vektor akan diklasifikasi kedalam 10 kelas menggunakan fungsi aktivasi Softmax.

Proses pelatihan dijalankan menggunakan GPU, sehingga waktu pelatihan yang dibutuhkan untuk melatih model CNN dalam 20 *epoch* yaitu 140 detik.

Proses pelatihan model CNN dilakukan dengan menggunakan 1750 data latih. Seperti yang disajikan pada Tabel II dan Gambar 5, dapat dilihat bahwa nilai galat dapat stabil turun mulai dari 1,545 pada *epoch* pertama sampai 0,0026 pada *epoch* terakhir yaitu *epoch* ke-20. Penurunan nilai galat terpantau tidak mengalami fluktuasi nilai ditengah-tengah proses pelatihan. Hal tersebut juga dapat dilihat pada nilai akurasi yang dicapai, mulai dari 51,77% pada *epoch* pertama sampai dengan 100% pada *epoch* terakhir. Model mampu mencapai nilai 100% pada epoch ke-11 dan pada titik tersebut sudah mulai konvergen sampai *epoch* terakhir yaitu *epoch* ke-20, dimana pada epoch terakhir yaitu *epoch* ke-20 mampu mendapatkan nilai akurasi 100% dan galat 0,0026.

B. Pengujian

Proses pengujian model CNN yang sudah melalui proses pelatihan dilakukan menggunakan 300 data uji yang berbeda dari data latih yang digunakan pada proses pelatihan. Model CNN yang terdiri dari bobot dan bias yang telah mampu menggeneralisir data latih dicoba untuk mengenali data uji yang berjumlah 300. Dapat dilihat pada Tabel III, bahwa pengujian yang dilakukan dengan model CNN diusulkan mampu menghasilkan nilai galat mencapai 0,017. Hal tersebut juga berbanding lurus dengan nilai akurasi yang didapat pada proses pengujian adalah 99,67%. Nilai akurasi yang didapat dari hasil pengujian model CNN dimampu mengungguli nilai akurasi pengujian pada penelitian-penelitian yang telah dilakukan sebelumnya. Hal ini dapat dilihat pada Tabel IV.

TABEL III HASIL PENGUJIAN

Pengujian	
Galat	Akurasi (%)
0,017	99,67

TABEL IV PERBANDINGAN AKURASI BEBERAPA METODE TSF DI INDONESIA

Perbandingan Akurasi	
Metode	Akurasi (%)
Shape Detection[9]	80,7
Template Matching[10]	88
Template Matching[12]	90,81
Klasifikasi CITY-BLOCK[11]	88
SVM dan KNN[13]	93,9
CHT[14]	61,3
CNN[14]	75
CNN[15]	97,33
CNN[16]	99,33
CNN (diusulkan)	99,67

IV. KESIMPULAN

Pengenalan rambu lalu lintas Indonesia menggunakan *Multi Scale Convolutional Network* telah disajikan. Pada proses pelatihan terhadap 1750 data latih dengan parameter pelatihan yaitu laju pelatihan 0,005, jumlah *epoch* 20, jumlah filter pada setiap lapisan konvolusi yaitu 32, didapatkan akurasi pelatihan sebesar 100% dimana nilai akurasi tersebut dapat dicapai mulai *epoch* ke-11 hingga *epoch* ke-20. Kemudian untuk proses pengujian, didapatkan nilai galat 0,017 dan nilai akurasi sebesar 99,67%. Hasil akurasi tersebut mampu mengungguli kajian-kajian sebelumnya tentang pengenalan rambu lalu lintas di Indonesia. Penelitian kedepannya dapat mengangkat penggunaan lebih dari satu masukan citra pada model CNN dan juga menggunakan algoritma pelatihan yang lebih baik dari SGD.

DAFTAR PUSTAKA

- [1] Kementerian Perhubungan, *Peraturan Menteri Perhubungan tentang Rambu Lalu Lintas Nomor 13 Tahun 2014*. 2014.
- [2] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems: Review and Future Perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6-22, 2014, doi: 10.1109/ITS.2014.2336271.
- [3] T. Günthner and H. Proff, "On the way to autonomous driving: How age influences the acceptance of driver assistance systems," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 81, pp. 586-607, Aug. 2021, doi: 10.1016/j.trf.2021.07.006.
- [4] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles," *IEEE Consum. Electron. Mag.*, vol. 7, no. 5, pp. 18-25, Sep. 2018, doi: 10.1109/MCE.2018.2828440.
- [5] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of advanced driver assistance systems (ADAS)," Thessaloniki, Greece, 2017, p. 120002. doi: 10.1063/1.5012394.
- [6] M. M. Narkhede and N. B. Chopade, "Review of Advanced Driver Assistance Systems and Their Applications for Collision Avoidance in

- Urban Driving Scenario,” in *Machine Learning and Big Data Analytics (Proceedings of International Conference on Machine Learning and Big Data Analytics (ICMLBDA) 2021)*, vol. 256, R. Misra, R. K. Shyamasundar, A. Chaturvedi, and R. Omer, Eds. Cham: Springer International Publishing, 2022, pp. 253–267. doi: 10.1007/978-3-030-82469-3_23.
- [7] 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.
- [8] A. Suriya Prakash, D. Vigneshwaran, R. Seenivasaga Ayyalu, and S. Jayanthi Sree, “Traffic Sign Recognition using Deep Learning for Autonomous Driverless Vehicles,” in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, Apr. 2021, pp. 1569–1572. doi: 10.1109/ICCMC51019.2021.9418437.
- [9] G. Romadhon and Murinto, “Aplikasi pengenalan citra rambu lalu lintas berbentuk lingkaran menggunakan metode jarak city-block,” *J. Sarj. Tek. Inform.*, vol. 2, no. 2, pp. 286–294, Jun. 2014.
- [10] O. R. Sitanggang, H. Fitriyah, and F. Utaminigrum, “Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Shape Detection Pada Raspberry Pi,” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 12, pp. 6108–6117, Dec. 2018.
- [11] T. Oddy Chrisdwianto, H. Fitriyah, and E. Rosana Widarsari, “Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode Template Matching,” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 3, pp. 1265–1274, Mar. 2018.
- [12] T. Harsono, A. Basuki, and N. Ramadijanti, “PENGENALAN GAMBAR RAMBU-RAMBU LALU LINTAS DENGAN METODE KUANTISASI RERATA,” *J. Math. Nat. Sci.*, vol. 16, no. 3, pp. 13–18, Sep. 2006.
- [13] C. Rahmad, I. Fauziah Rahmah, and R. Andrie Asmara, “Deteksi dan pengenalan rambu lalu lintas di indonesia menggunakan RGBN dan Gabor,” in *SENTRINOV*, 2017, vol. 3, pp. TII3-22.
- [14] A. Triyadi and F. Utaminigrum, “Pengembangan Sistem Rekognisi Rambu Kecepatan Menggunakan Circle Hough Transform dan Convolutional Neural Network Berbasis Raspberry Pi,” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 1, pp. 56–64, Jan. 2020.
- [15] M. Akbar, “Traffic sign recognition using convolutional neural networks,” *J. Teknol. Dan Sist. Komput.*, vol. 9, no. 2, pp. 120–125, Apr. 2021, doi: 10.14710/jtsiskom.2021.13959.
- [16] M. Akbar, “Pengenalan Rambu Lalu-lintas menggunakan Convolutional Neural Network (Studi Kasus: Rambu Lalu-lintas Indonesia),” *J. Nas. Inform. Dan Teknol. Jar.*, vol. 6, no. 2, pp. 272–276, Mar. 2022, doi: <http://dx.doi.org/10.30743/infotekjar.v6i2.4564>.
- [17] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, “Object Recognition with Gradient-Based Learning,” in *Shape, Contour and Grouping in Computer Vision*, vol. 1681, Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. doi: 10.1007/3-540-46805-6_19.
- [18] P. Sermanet and Y. LeCun, “Traffic sign recognition with multi-scale Convolutional Networks,” in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, Jul. 2011, pp. 2809–2813. doi: 10.1109/IJCNN.2011.6033589.
- [19] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
- [20] A. P. Engelbrecht, *Computational intelligence: an introduction*, 2nd ed. Chichester, England; Hoboken, NJ: John Wiley & Sons, 2007.
- [21] S. W. Smith, *The scientist and engineer’s guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1999.
- [22] 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.
- [23] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning.” arXiv, Nov. 08, 2018. Accessed: Aug. 06, 2022. [Online]. Available: <http://arxiv.org/abs/1811.03378>
- [24] B. Zhou, C. Han, and T. Guo, “Convergence of Stochastic Gradient Descent in Deep Neural Network,” *Acta Math. Appl. Sin. Engl. Ser.*, vol. 37, no. 1, pp. 126–136, Jan. 2021, doi: 10.1007/s10255-021-0991-2.