

# Pengenalan Pola Lintasan Berbasis *Neural Network* Pada *Prototype Self-Driving Car*

Leonardo Rudolf Manangka<sup>1)</sup>, Herwin Suprijono<sup>2)</sup>, Dedi Nurcipto<sup>3)</sup>

<sup>1, 2, 3)</sup> Program Studi Teknik Elektro, Fakultas Teknik Universitas Dian Nuswantoro Semarang

e-mail: [511201500700@mhs.dinus.ac.id](mailto:511201500700@mhs.dinus.ac.id)<sup>1)</sup>, [herwin.suprijono@dsn.dinus.ac.id](mailto:herwin.suprijono@dsn.dinus.ac.id)<sup>2)</sup>, [dedi.nurcipto@dsn.dinus.ac.id](mailto:dedi.nurcipto@dsn.dinus.ac.id)<sup>3)</sup>

## ABSTRACT

*Self-driving cars are an interesting topic to discuss due to the high level of traffic accidents that occur due to human error. Self driving cars are vehicles that can find out about the environment with minimal human intervention. Self driving itself has many development methods such as Light Detection and Ranging (LIDAR), cameras, radars, or a combination of these sensors. This study made a prototype self-driving car using a camera as a sensor and a neural network algorithm for pattern recognition. The pattern recognition in question is the image recognition of the path taken. The data that has been taken will later be converted into a matrix with dimensions of 320x120 according to the image resolution. Then the data will be trained to recognize the path pattern with the proportion of 7: 3 for training accuracy and validation accuracy. The resulting prediction has an accuracy of 76.86% for training accuracy and 75.24% for validation accuracy.*

**Keywords:** *Neural Network, Self-Driving Car, Camera*

## ABSTRAK

*Self driving car* menjadi topik yang menarik untuk dibahas karena tingginya tingkat kecelakaan lalu lintas yang terjadi akibat *human error*. *Self-driving car* merupakan kendaraan yang dapat mengetahui lingkungan dengan sedikit campur tangan manusia. *Self-driving* sendiri memiliki banyak metode pengembangan seperti *Light Detection and Ranging (LIDAR)*, kamera, radar, atau kombinasi dari sensor tersebut. Penelitian ini membuat *prototype self-driving car* dengan menggunakan kamera sebagai sensor dan algoritma jaringan saraf tiruan untuk pengenalan pola. Pengenalan pola yang dimaksud adalah pengenalan gambar dari jalur yang diambil. Data yang telah diambil nantinya akan diubah menjadi matriks dengan dimensi 320x120 sesuai resolusi gambar. Kemudian data akan dilatih untuk mengenali pola jalur dengan proporsi 7: 3 untuk akurasi pelatihan dan akurasi validasi. Prediksi yang dihasilkan memiliki akurasi 76,86% untuk akurasi pelatihan dan 75,24% untuk akurasi validasi.

**Kata kunci:** Jaringan Syaraf Tiruan, *Self-Driving Car*, Kamera

## I. PENDAHULUAN

**S***elf driving cars* menjadi topik yang menarik untuk dibahas dikarenakan tingkat kecelakaan lalu lintas yang banyak terjadi karena faktor kesalahan manusia. Angka kecelakaan lalu lintas yang disebabkan oleh faktor kesalahan manusia mencapai 61%, faktor kendaraan 9%, faktor prasarana dan lingkungan 30%[1]. Karena tingginya kecelakaan lalu lintas diakibatkan oleh faktor kesalahan manusia, maka penelitian tentang *self driving* diperlukan untuk membantu mengurangi angka kecelakaan yang disebabkan oleh faktor kesalahan manusia.

*Self driving cars* adalah sebuah kendaraan yang dapat mengetahui lingkungan sekitar dengan sedikit campur tangan manusia. *Self driving* sendiri memiliki banyak metode pengembangan seperti *Light Detection and Ranging (LIDAR)*, kamera, radar, ataupun kombinasi dari sensor tersebut[2]. Penggunaan sensor tersebut bertujuan untuk mengenali lingkungan sekitar seperti *mark* jalan dan rambu-rambu lalu lintas.

Penelitian ini akan menggunakan *prototype* dengan 4 motor DC sebagai aktuator, kamera sebagai sensor, Raspberry Pi sebagai pemrosesan gambar dan komputer sebagai pemrosesan *Neural Network (NN)*. *NN* adalah paradigma pengolahan informasi yang terinspirasi oleh

sistem saraf biologis, seperti proses informasi otak manusia. Struktur dari sistem pengolahan informasi ini terdiri dari sejumlah besar pemrosesan yang saling berhubungan (*neuron*) yang bekerja serentak untuk menyelesaikan masalah tertentu. *NN* bekerja seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah *NN* dikonfigurasi untuk aplikasi tertentu seperti pengenalan pola, klasifikasi data, melalui proses pembelajaran[3]. Penelitian ini ingin membangun *prototype self driving cars* dengan menerapkan metode *NN* yang digunakan untuk mengenali lingkungan sekitar seperti arah jalan.

Agar penelitian ini dapat dilakukan lebih fokus, maka perlu dibatasi variabelnya. Pada penelitian ini *prototype self-driving car* dibatasi hanya pada arah pergerakannya saja. Seperti pengenalan pola jalur lurus, pola garis belok kanan dan kiri.

## II. TEORI DASAR

### A. *Self-Driving*

*Self driving* merupakan salah satu cara mengendalikan mobil dengan sedikit campur tangan manusia. *Self driving* menggunakan sensor sebagai cara untuk mengenali lingkungan sekitar. Pengenalan tersebut berupa *mark* jalan, rambu-rambu lalu lintas,

mobil, dan juga pejalan kaki. Sensor yang digunakan antara lain dapat berupa LIDAR, kamera, GPS, radar atau gabungan dari sensor tersebut[2].

Self driving menurut *Society of Automotive Engineers* (SAE) memiliki 6 tingkatan.

- Level 0, semua sistem dikendalikan oleh manusia
- Level 1, beberapa sistem seperti kecepatan dan rem otomatis dikendalikan oleh mobil satu per satu.
- Level 2, mobil dapat menawarkan opsi seperti *acceleration and steering* tapi membutuhkan manusia untuk beroperasi dengan aman.
- Level 3, mobil dapat mengatur keamanan pada kondisi tertentu, tapi pengemudi dapat mengambil alih ketika diberi notifikasi.
- Level 4, mobil dapat beroperasi secara otomatis, meski tidak semuanya.
- Level 5, *self driving* diterapkan secara menyeluruh dikondisi apapun.

Dari beberapa level diatas level 0 – 2 pengemudi memonitor lingkungan mengemudi, sedangkan level 3 – 5 *self driving* memonitor lingkungan mengemudi[9].

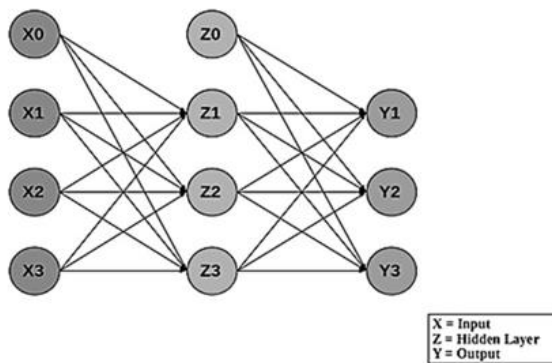
### B. Neural Network

NN merupakan pengolahan informasi yang terinspirasi dari sistem saraf biologis manusia.

Pengolahan informasi ini terdiri dari jumlah pemrosesan yang saling terhubung atau disebut dengan *neuron*, yang berkerja untuk menyelesaikan masalah tertentu. *Neural network* bekerja seperti jaringan saraf manusia melalui tahap pembelajaran berupa contoh-contoh data yang diberikan. Misalnya untuk mengenali pola atau klasifikasi data melalui tahap pembelajaran.

Gambar 1. Arsitektur Neural Network

merupakan arsitektur NN.

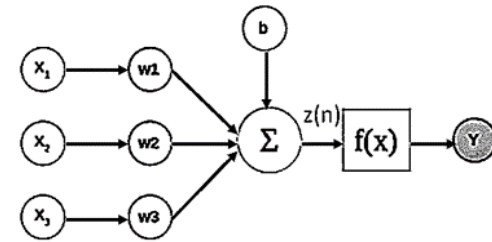


Gambar 1. Arsitektur Neural Network

NN memiliki arsitektur yang tersusun menjadi tiga lapisan antara lain, *input layer*, *hidden layer* dan *output layer*. *Input layer* merupakan lapisan yang bertugas menerima inputan dari luar seperti data sensor. *Hidden layer* merupakan lapisan yang dimana nilai bobot dan fungsi aktivasi berada. *Output layer* merupakan hasil

pemecahan suatu masalah. Gambar 2 merupakan model matematis NN.

Gambar 2. Model matematis Neural Network



$$z(n) = (\sum_{i=1}^n w_i + x_i) + b \quad (1)$$

Dimana :  $z(n)$  = hasil dari jumlah input

$w$  = weight

$x$  = nilai input

Setelah didapatkan nilai  $z(n)$  kemudian diteruskan ke fungsi aktivasi tertentu  $f(X)$  dan didapat *output neuron* adalah Y.

*Backpropagation* merupakan mekanisme utama dimana NN belajar. *Backpropagation* memiliki tiga tahap yang harus dilakukan dalam tahap pelatihan. Pertama adalah perambatan maju (*forward propagation*), kedua tahap perambatan balik dan terakhir adalah perubahan bobot dan bias. Arsitektur ini terdiri dari *input layer*, *hidden layer* dan *output layer*.

Kelebihan-kelebihan yang diberikan oleh neural network antara lain:

1. Belajar *adaptive*: Kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organisation*: Sebuah NN dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar.
3. *Real Time Operation*: Perhitungan NN dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini[3].

### III. METODE PENELITIAN

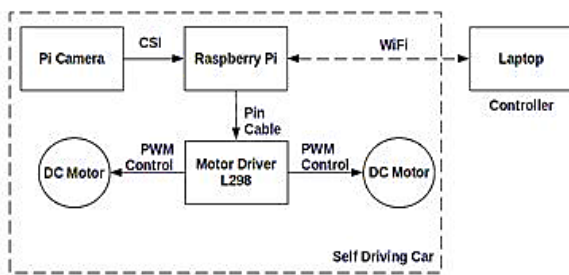
Tahap awal penelitian ini dilakukan pengumpulan informasi dan refrensi dari penelitian terdahulu. Beberapa hal yang dibutuhkan pada penelitian ini yaitu pembuatan alur penelitian, perancangan *hardware*, dan perancangan *software*. Kemudian dilakukan pengumpulan data, pelatihan *self driving car*, dan tahap terakhir adalah pengujian *self driving car*.

#### A. Perancangan Hardware

Perancangan *hardware* pada penelitian ini menggunakan *Raspberry Pi* yang terhubung dengan sensor yaitu berupa kamera, serta *Raspberry Pi* yang terhubung dengan *driver motor* dimana *driver motor* menggerakkan motor DC. Mini komputer *Raspberry Pi*

bertugas untuk mengambil gambar yang nantinya akan dikirim ke laptop secara nirkabel. Gambar 3 merupakan diagram blok sistem.

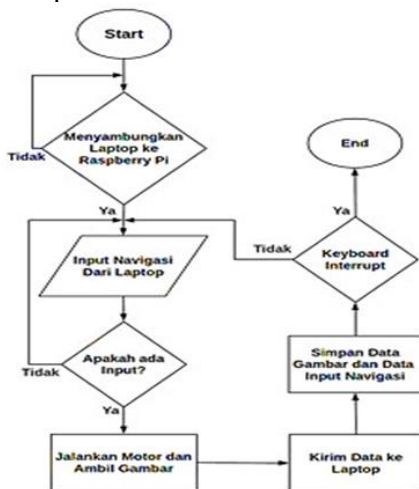
Gambar 3. Diagram blok sistem



Laptop berfungsi sebagai *controller*, pengolahan *neural network*, penerima data gambar dari *Raspberry Pi* dan menyimpan data gambar serta mengirim hasil proses *neural network* kembali ke *Raspberry Pi*. *Raspberry Pi* berfungsi untuk mengambil gambar dengan menggunakan kamera, mengirim data gambar ke laptop dan mengirim sinyal *Pulse Width Modulation* (PWM) ke rangkaian *driver motor* untuk menjalankan motor. Laptop dan *Raspberry Pi* menggunakan jaringan *WiFi* untuk berkomunikasi.

### B. Pengumpulan Data

Tahap pengumpulan data dilakukan dengan caramenjalankan *self driving car* secara manual, dimana gambar diambil ketika mobil berjalan, menggunakan laptop sebagai pengendali secara nirkabel. Data yang diambil berupa gambar jalan dengan resolusi 320 x 240 *pixel*. Input navigasi yang dikirim dari laptop berguna sebagai label yang nantinya juga akan disimpan sebagai klasifikasi pengenalan gambar. Data yang diambil nantinya akan disimpan di laptop sebagai data *set training NN* yang berfungsi sebagai data latih algoritma *NN*. Berikut ini adalah gambar *flowchart* alur program dapat dilihat pada Gambar 4.



Gambar 4. Flowchart alur program pengumpulan data

### C. Pelatihan

Tahap pelatihan, dilakukan dengan mengkonversi gambar ke bentuk *array* yang merupakan suatu tipe data terstruktur yang dapat menyimpan banyak data dengan suatu nama yang sama dan menempati tempat di memori yang berurutan serta bertipe data sama. Kemudian data

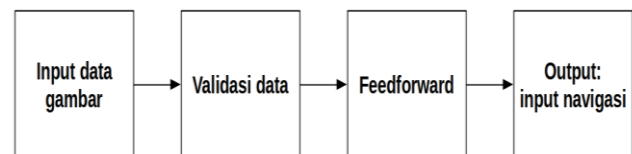


gambar yang sudah dikonversi akan dipasangkan dengan data label yang sudah disimpan. Setelah itu data akan dilatih dengan menggunakan metode *backpropagation*. Struktur *NN* memiliki input *layer* sebanyak data *array* yang disimpan, 1 *hidden layer*, 4 *output layer* yang berupa input navigasi seperti maju, mundur, kanan dan kiri. Kemudian setiap perubahan pada bobot akan disimpan, yang nantinya akan digunakan sebagai data acuan pada saat pengujian. Gambar 5 merupakan gambar diagram blok pelatihan

Gambar 5. Diagram blok pelatihan

### D. Pengujian

Tahap pengujian, dilakukan dengan menerapkan hasil pelatihan dengan jalur lintasan. Pada tahap ini *self driving car* akan mengambil data dari lingkungan yang baru kemudian validasi data dengan data bobot yang tersimpan sebagai



acuan. Gambar 6 merupakan gambar diagram blok pengujian.

Gambar 6. Diagram blok pengujian

Setelah proses *feedforward* akan dihasilkan output berupa input navigasi untuk menentukan arah gerakan dari mobil. Input navigasi ini yang akan dikirim ke mikrokontroler sebagai perintah yang akan mengatur kecepatan motor.

### E. Analisis

Pada tahap analisis akan dilakukan pengujian pada sistem *self driving*. Pengujian ini berupa menguji kepresisian sistem dalam menangani lintasan yang dibuat. Akan ada 4 lintasan uji yang akan menguji tingkat kepresisian *self-driving*. Lintasan uji pertama memiliki bentuk melingkar dengan satu arah belokan dan pola garis berwarna hitam, lintasan uji kedua memiliki bentuk melingkar dengan lintasan yang lebih rumit dan pola garis berwarna hitam, lintasan uji ketiga memiliki bentuk persegi panjang dan pola garis berwarna hitam, dan lintasan keempat memiliki pola garis berwarna putih.





### C. Pelatihan Model Neural Network

Setelah dilakukan pengumpulan dan *preprocessing* data, selanjutnya data dimasukkan ke dalam model *neural network*.

Pelatihan model *neural network* yang digunakan adalah *backpropagation* dengan inputan berupa data array yang diambil dari data *train.npy* yang sudah di normalisasi. Hasil pelatihan dapat dilihat pada Gambar 13

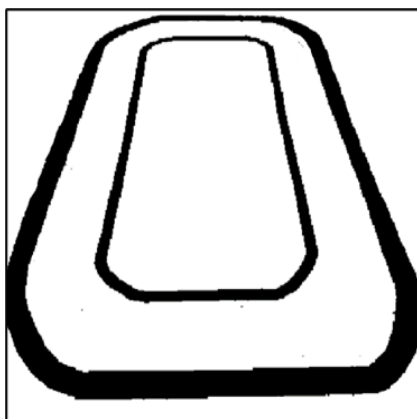
```
image (347, 38400)
labels (347, 4)
Training ...
Training duration: 99.98s
Train accuracy: 76.86%
Validation accuracy: 75.24%
Model saved to: 'saved_model/nn_model.xml'
```

Gambar 13. Hasil training model

Model *NN* yang dirancang menggunakan input layer sebanyak 38400 angka ini didapatkan dari resolusi gambar yaitu  $320 \times (240/2)$ , satu *hidden layer*, dan empat *output layers*. Iterasi untuk melatih *NN* sebanyak 100 kali dengan *learning rate* sebesar 0,01. Model *NN* ini menggunakan 347 buah citra dan dikonversi menjadi array menggunakan modul Numpy. *Dataset* kemudian dipisah dengan perbandingan 7:3 dimana 7 adalah data untuk *training* dan 3 adalah data *test*. Dari hasil pelatihan model *neural network* didapatkan *training accuracy* sebesar 76.86% dan *validation accuracy* 75.24%. Bobot dari *training* ini akan di simpan dengan nama file *nn\_model.xml*. Data bobot yang disimpan nantinya akan digunakan untuk pengujian dan implementasi *self-driving car*. Data tersebut menunjukkan nilai bobot yang didapat dari hasil *training*, nilai ini yang akan dipakai untuk menentukan hasil prediksi dalam tahap pengujian dan implementasi.

### D. Pengujian dan Implementasi self-driving car

Pengujian dan implementasi *NN* ke *prototype self-driving car* dilakukan dengan 4 lintasan berbeda. Lintasan pertama menggunakan lintasan melingkar dan lintasan kedua dilakukan dengan menambahkan berhasil dilakukan oleh *self-driving car* dengan maksimal 10 putaran. Bentuk lintasan ke-1 dapat dilihat pada **Error! Reference source not found.**



Gambar 14. Bentuk lintasan uji ke-1

Pada pengujian dengan lintasan uji ke-1 dilakukan 10 kali percobaan putaran searah jarum jam (*clockwise*) dalam 10 kali putaran didapati tingkat keberhasilan sebesar 100% dan pada 10 kali putaran berlawanan arah jarum jam (*counter clockwise*) didapati tingkat keberhasilan sebesar 100%.

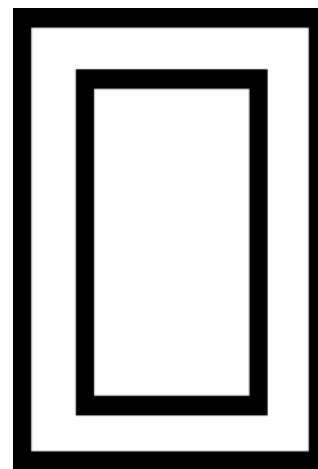
Untuk pengujian berikutnya dilakukan pada pola lintasan yang lebih kompleks bentuk pola lintasan dapat dilihat pada Gambar 16.



Gambar 16. Bentuk lintasan uji ke-2

Dengan pola lintasan yang lebih kompleks dibandingkan pola lintasan ke-1 setelah diujicoba dengan cara yang sama dengan uji ke-1 maka *NN* dapat beradaptasi dengan perubahan pola lintasan yang baru dimana dalam 10 kali putaran uji coba didapati tingkat keberhasilan sebesar 100% dan 10 kali putaran berlawanan arah jarum jam (*counter clockwise*) didapati tingkat keberhasilan sebesar 100%.

Kemudian akan dilakukan pengujian ke-3 dengan tikungan sudut  $90^\circ$  seperti pada Gambar 17. Dari hasil pengujian didapat keakuratan pengujian sebesar 50%. Terjadi kesalahan *prototype self-driving car* tidak mampu berbelok tetapi justru berjalan lurus.



Gambar 17. Bentuk lintasan uji ke-2

Pengujian ke-4 dilakukan dengan mengganti pola garis berwarna hitam menjadi pola garis berwarna putih. Tujuan pengujian ini untuk mengetahui seberapa baik

*prototype self-driving car* dapat memprediksi jalur dengan pola tersebut. Hasil pengujian menunjukkan *Prototype self-driving car* gagal dalam mengenali pola lintasan berwarna putih disebabkan karena tidak adanya *dataset* atau pembelajaran pengenalan citra untuk pola lintasan berwarna putih.

## V. KESIMPULAN

Keakuratan dari prediksi dari *NN* dalam pengenalan pola lintasan tergantung dari jumlah pola yang dilatih, dari hasil pengujian dengan lintasan ke-1 dan ke-2 menunjukkan keakuratan yang baik karena *NN* mendapat pembelajaran pola pada kedua lintasan tersebut. Sedangkan untuk percobaan dengan pola lintasan ke -3 dan ke -4 dengan keakuratan *NN* dalam memprediksi menurun dimana *NN* tidak dapat mengenal citra pola lintasan yang diberikan hal ini disebabkan *NN* tidak mendapat dataset pelatihan belokan dengan sudut 90° dan pola lintasan dengan garis putih.

Untuk meningkatkan keakuratan prediksi dari *NN* dengan memberikan citra pola-pola yang lebih lengkap dan iterasi pelatihan yang lebih banyak sehingga *training accuracy* dan *validation accuracy* semakin baik, tentu nya dengan menggunakan komputasi yang lebih cepat.

Pengembangan berikutnya pada *Prototype self-driving car* dapat mengenali halangan misalkan kendaraan lain, halangan di lintasan, rambu-rambu lalu lintas, menanamkan *embedded system* yang lebih cepat dan menambahkan sistem kendali *Prototype self-driving car*.

## DAFTAR PUSTAKA

- [1] K. K. dan Informatika, "Setiap Jam Rata-Rata 3 Orang Meninggal Akibat Kejelakan Jalan Di Indonesia," 2017. [https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel\\_gpr](https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr).
- [2] Sumardi, M. Taufiqurrahman, and M. A. Riyadi, "Street mark detection using raspberry pi for self-driving system," *Telkonnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 2, pp. 629–634, 2018, doi: 10.12928/TELKOMNIKA.v16i2.4509.
- [3] S. K. Kokate, "Review on Autonomous Car using Raspberry Pi," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 1, pp. 3090–3094, 2018, doi: 10.22214/ijra-set.2018.1427.
- [4] G. SinghPannu, M. Dawud Ansari, and P. Gupta, "Design and Implementation of Autonomous Car using Raspberry Pi," *Int. J. Comput. Appl.*, vol. 113, no. 9, pp. 22–29, 2015, doi: 10.5120/19854-1789.
- [5] K. McFall, "Using visual lane detection to control steering in a self-driving vehicle," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 166, pp. 861–873, 2016, doi: 10.1007/978-3-319-33681-7\_77.

- [6] S. Jana, S. Borkar, and M. E. Student, "Autonomous Object Detection and Tracking using Raspberry Pi," *Int. J. Eng. Sci. Comput.*, vol. 7, no. 7, pp. 14151–14155, 2017, [Online]. Available: <http://ijesc.org/>.
- [7] W. Farag and Z. Saleh, "Traffic signs identification by deep learning for autonomous driving," *IET Conf. Publ.*, vol. 2018, no. CP747, pp. 22–23, 2018, doi: 10.1049/cp.2018.1382.
- [8] M. G. Bechtel, E. McEllhiney, M. Kim, and H. Yun, "DeepPicar: A low-cost deep neural network-based autonomous car," *Proc. - 2018 IEEE 24th Int. Conf. Embed. Real-Time Comput. Syst. Appl. RTCSA 2018*, pp. 11–21, 2019, doi: 10.1109/RTCSA.2018.00011.
- [9] Global Ground Vehicle Standards, "Automated Driving: Levels Of Driving Automation Are Defined In New SAE International Standard J3016," *SAE International*, 2016. [https://www.sae.org/misc/pdfs/automated\\_driving.pdf](https://www.sae.org/misc/pdfs/automated_driving.pdf).
- [10] A. Driving and S. Dimensions, "Autonomes Fahren," *Auton. Fahr.*, 2015, doi: 10.1007/978-3-662-45854-9.