

Terbit online pada laman web jurnal: <https://jurnal.plb.ac.id/index.php/tematik/index>

T E M A T I K

Jurnal Teknologi Informasi Komunikasi (e-Journal)

Vol. 10 No. 2 (2023) 264 - 269

ISSN Media Elektronik: 2443-3640

Penghindaran Rintangan Otomatis Pada Agen Otonom Berbasis End-to-End Deep Imitation Learning

Automatic Obstacle Avoidance in Autonomous Agents Based on End-to-End Deep Imitation Learning

Handoko Supeno¹, Muhammad Tirta Mulia², Mellia Liyanthy³, Kevin Anggara Putra⁴, Fauzan Nursalma Mawardi⁵^{1,2,3,4,5}Teknik Informatika, Fakultas Teknik, Universitas Pasundan¹handoko@unpas.ac.id, ²tirta.mulia@unpas.ac.id, ³liyanthy@unpas.ac.id, ⁴193040086@mail.unpas.ac.id, ⁵193040053@mail.unpas.ac.id

Abstract

This research addresses the issue of automatic obstacle avoidance during navigation by autonomous agents. Designing a traditional programmed automatic obstacle avoidance system would be difficult and expensive. Therefore, a neural network-based approach is proposed, known as end-to-end Deep Imitation Learning, where the approach is data-driven and thus relatively easier and more cost-effective compared to traditional methods. The research also proposes the architecture of a convolutional neural network design and image processing techniques for effective and efficient machine learning training. Testing is conducted on a path with randomly placed obstacles in the Webots simulator. Gradual performance evaluations demonstrate that the proposed architecture successfully trains autonomous agents to maneuver when encountering dynamic obstacles with a relatively small training dataset.

Keywords: Autonomous agent, end-to-end deep imitation learning, collision avoidance

Abstrak

Penelitian ini mengangkat isu mengenai penghindaran rintangan otomatis pada saat navigasi oleh agen otonom. Perancangan sistem penghindaran rintangan otomatis secara pemrograman tradisional akan sulit dan mahal. Oleh karena itu diusulkan pendekatan berbasis jaringan syaraf dalam secara langsung dari masukan ke kontrol yang dikenal sebagai *end-to-end Deep Imitation Learning*, dimana pendekatan ini akan berbasis data pakar sehingga relatif lebih mudah dan murah dibandingkan dengan pendekatan tradisional. Penelitian ini juga mengusulkan arsitektur desain jaringan syaraf konvolusi serta teknik pemrosesan citra yang mampu menghasilkan pelatihan pembelajaran mesin dalam yang efektif dan efisien. Pengujian dilakukan pada jalur dengan rintangan yang dipasang secara acak pada simulator *Webots*. Evaluasi performa secara bertahap membuktikan bahwa arsitektur yang diusulkan berhasil melatih agen otonom untuk dapat bermanuver ketika menemui rintangan dinamis dengan jumlah dataset pelatihan yang relatif sedikit.

Kata kunci : Agen otonom, end-to-end deep imitation learning, penghindaran rintangan.

1. Pendahuluan

Agen otonom merupakan robot atau kendaraan yang dalam menjalankan fungsinya tidak dikendalikan oleh manusia melainkan bergerak berdasarkan keputusan agen itu sendiri. Untuk mencapai kemampuan ini, kendaraan otonom biasanya dipasang beberapa sensor seperti kamera, lidar, radar, dsb. Sensor-sensor tersebut dipasang sedemikian rupa dengan arsitektur yang dikolaborasi dengan perangkat komputasi tertentu. Pengembangan agen otonom di masa depan sangatlah

dibutuhkan pada bidang seperti militer [1], kemanusiaan [2], dan komersil[3].

Salah satu kemampuan yang paling diperlukan oleh agen otonom adalah kemampuan navigasi untuk dapat bergerak dari satu titik lokasi ke titik yang lainnya secara mandiri. Namun dalam proses navigasi ini, kadang agen menghadapi beragam rintangan yang harus dilewati sehingga agen harus bermanuver agar tetap bisa mencapai tujuan. Permasalahannya adalah bagaimana jika agen ditempatkan di lingkungan dimana

posisi rintangan tersebut tidak pasti, untuk menyelesaikan hal ini maka agen harus memiliki kemampuan untuk mengenali rintangan dimanapun dan kapanpun rintangan tersebut dihadapi dan kemampuan untuk merespon rintangan dengan kemampuan manuver yang baik. Kemampuan untuk mengenali dan melakukan manuver tersebut dinamakan sebagai kemampuan penghindaran rintangan.

Penelitian ini menawarkan sebuah pendekatan pelatihan berbasis kecerdasan buatan yang memanfaatkan Jaringan syaraf dalam artifisial (*Deep Neural Network / DNN*) dengan paradigma pembelajaran imitasi untuk menghasilkan agen yang dapat melakukan penghindaran rintangan dinamis secara otomatis. DNN yang dihasilkan lebih tepatnya varian Jaringan Syaraf Konvolusi (*Convolutional Neural Network/CNN*) yang memiliki kemampuan untuk melakukan ekstraksi fitur dari gambar yang kemudian bisa dihubungkan dengan kontrol. Fitur tersebut dapat merepresentasikan kondisi lingkungan agen, termasuk keberadaan rintangan didepannya yang dapat menjadi dasar keputusan kontrol seperti pada penelitian [4] dan [5].

Selanjutnya peneliti tertarik untuk mengusulkan arsitektur desain jaringan syaraf konvolusi serta teknik pemrosesan citra yang mampu menghasilkan pelatihan pembelajaran mesin dalam yang efektif dan efisien. Kontribusi pada penelitian ini antara lain bertujuan untuk menghasilkan:

- (1) Mekanisme pelatihan agen dengan paradigma *end-to-end* yaitu sebuah pendekatan implementasi dimana agen menangkap situasi sekitar dari sensor kamera yang terpasang lalu merespon dengan manuver jika terdeteksi rintangan serta bagaimana mengolah citra masukan dengan teknik pemrosesan citra;
- (2) Desain CNN yang dapat mendukung pendekatan *end-to-end* pada poin (1);
- (3) Tahapan pelatihan agen untuk poin(1) dan (2) sesuai kaidah pembelajaran mesin yaitu penyiapan dataset, pelabelan, pelatihan, dan evaluasinya.

2. Metode Penelitian

Kapabilitas agen otonom yang dibutuhkan untuk navigasi dan penghindaran rintangan, paling tidak harus dapat meneruskan masukan sensor kamera sebagai persepsi ke kontrol, memiliki desain jaringan syaraf dalam yang sesuai, dan mekanisme pelatihan untuk dapat mengenalkan rintangan dan manuvernya. Pada bab ini akan dijelaskan komponen-komponen tersebut pada subbab berikut.

2.1. Persepsi ke kontrol pada agen otonom *End-to-End*

Penelitian ini tidak menggunakan pendekatan modular atau *pipeline* dimana perilaku agen otonom dibangun atas berbagai algoritma pada aktivitas mengemudi dengan konsep modular untuk persepsi, lokalisasi,

prediksi perilaku berkendara, pengambilan keputusan, kontrol pergerakan, dsb. Penyelesaian tugas-tugas inilah yang lalu disatukan agar dapat mengemudikan kendaraan secara otomatis [6] sebagaimana yang diperlihatkan pada Gambar 1. Karena begitu banyaknya ketidakpastian pada lingkungan operasional di lapangan maka ada tuntutan beradaptasi terhadap skenario baru yang berdampak kepada perancangan dan pengembangan solusi manual untuk tiap skenario dan tugas yang dihadapi. Pendekatan ini dianggap tidak praktis [7].



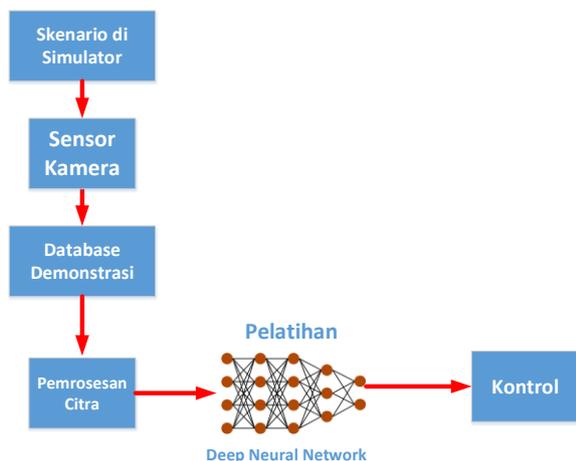
Gambar 1. Pendekatan Modular(Pipeline) Untuk Agen Otonom

Oleh karena pertimbangan diatas, maka agen otonom pada penelitian ini dibangun menggunakan pendekatan *end-to-end* dimana pendekatan yang menyelesaikan persepsi, keputusan, dan kontrol secara simultan dan tanpa memerlukan rekayasa perangkat lunak secara manual [8], lebih spesifik pendekatan *end-to-end* berbasis DNN yang digabungkan dengan konsep *Imitation Learning* sehingga dikenal sebagai *Deep Imitation Learning(DIL)*. *Imitation Learning* adalah sebuah pendekatan pembelajaran dimana agen mencoba meniru manusia, konsepnya mirip dengan pembelajaran tersupervised yang lazim digunakan pada pembelajaran mesin. Dari pencarian yang dilakukan, nampaknya beragam pendekatan penghindaran rintangan untuk agen otonom pernah diusulkan oleh peneliti tanah air, namun sepengetahuan penulis masih belum ada penelitian lokal yang mengangkat penghindaran rintangan berbasis *End-to-End Deep Imitation Learning*. Maka terdapat prospek untuk membangun arsitektur yang mampu untuk melatih agen otonom *end-to-end* berbasis DIL.

Pada konteks kemampuan persepsi, sebuah agen otonom harus dipasang sebuah sensor kamera yang dapat menangkap gambar, lalu dilakukan pemrosesan citra yang terdiri atas kompresi citra, penyesuaian kanal warna dari BGRA menjadi RGB, kompresi kanal warna dari 3 dimensi menjadi 1 dimensi, dan normalisasi nilai piksel yang awalnya memiliki nilai dari 0 sd 255 menjadi 0 sd 1. Citra yang sudah di proses ini akan diberikan pada jaringan syaraf dalam yang kemudian akan menghasilkan kontrol setir untuk manuver penghindaran rintangan. Gambaran persepsi ke kontrol serta hubungannya dengan arsitektur pelatihan ditunjukkan pada Gambar 2.

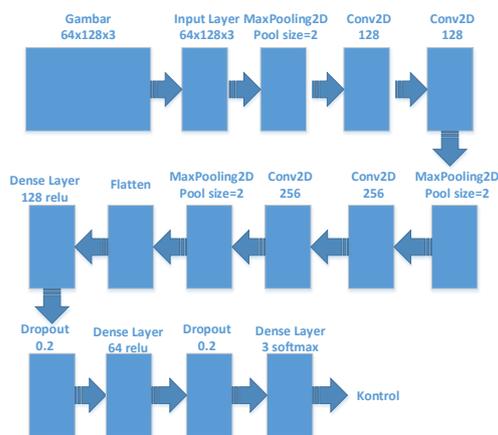
Satu isu terkait dengan kendaraan otonom, yaitu jarang sekali pelatihan dibidang ini dilakukan secara langsung

dilapangan. Pelatihan secara langsung memiliki resiko dan biaya yang tinggi. Oleh karena itu lazim penelitian menggunakan simulator untuk melatih kendaraan otonomnya seperti [9]. Simulator yang lazim digunakan pada bidang kendaraan otonom antara lain CARLA (Car Learning to Act) [10], AirSim[11] , SUMO (Simulation of Urban Mobility) [12], TORCS (The Open Racing Car Simulator)[13], dan WEBOTS[14] yang digunakan pada penelitian ini. Baik lingkungan, sensor, dan agen otonom yang dimaksud disimulasikan pada simulator Webots ini.



Gambar 2. Arsitektur *End-to-End* Untuk Agen Otonom berbasis DIL
 2.2. Desain Jaringan Syaraf Dalam

Arsitektur jaringan syaraf dalam yang digunakan adalah *Convolutional Neural Network* (CNN). CNN dapat dibagi menjadi 2 komponen utama, pertama adalah *Feature Learning* yang melakukan pengkodean dari sebuah gambar menjadi fitur-fitur yang merepresentasikan gambar tersebut (*Feature Extraction*) dan kedua adalah *Fully-Connected Layer* untuk mengeluarkan kontrol. Pendekatan ini terinspirasi dari beberapa penelitian, salah satunya adalah [15].



Gambar 3. Arsitektur CNN

Desain arsitektur CNN yang digunakan pada penelitian ini terdiri atas *input layer*, *max pooling layer*, *Conv2D*, *Dense Layer*, dan *Dropout* yang disusun pada Gambar 3. Bagian *Feature Extraction* diberikan masukan berupa citra yang kemudian dilakukan konvolusi oleh kernel menganalisa struktur data gambar yang kemudian akan menghasilkan *feature map* dimana data masukan $x(t)$ dan *kernel* $w(t)$ digunakan untuk memetakan $x(t)$ ke *feature-map* $s(t)$. Tujuan pemetaan ini adalah untuk mereduksi ukuran dari parameter pada layer-layer berikutnya. Ekspresi matematis dari konvolusi dapat dituliskan pada Persamaan 1.

$$s(t) = (x * w)(t) \tag{1}$$

Selanjutnya pada bagian *Fully Connected Layer* pada dasarnya menggunakan struktur *Neural Network*(NN) biasa. Dari layer konvolusi yang kemudian dilakukan perubahan bentuk menjadi satu dimensi dalam proses *flatten*, NN memodelkan fungsi $f(x, \theta)$ dengan menyetel parameter θ dimana x adalah fitur dari bagian *Feature Extraction* sebelumnya. Hubungan antar node di jaringan syaraf bagian ini dapat didefinisikan seperti terlihat pada Persamaan 2.

$$a_j^i = \phi \left(\sum_{k=0}^N w_{jk}^i x_{jk}^i + b_j^i \right), w_{jk}^i, b_j^i \in \theta \tag{2}$$

ϕ adalah transformasi non-linier, N adalah jumlah *node* masukan, b adalah bias, w adalah bobot (*weight*) dan x adalah masukan.

2.3. Kerangka Kerja Pelatihan

Pada sub bab ini akan dijabarkan mengenai tahapan dalam membangun agen otonom berbasis DIL. Tahapan kerja yang dilakukan secara garis besar antara lain persiapan lingkungan simulasi, pengumpulan dataset yang dimasukkan ke database demonstrasi, untuk kemudian dilakukan pelatihan.

Pada tahapan persiapan lingkungan simulasi terdapat beberapa aktivitas antara lain adalah melakukan pengaturan sensor kamera yang terdiri atas resolusi dan frame per second yang digunakan, mempersiapkan lingkungan simulasi yang ditambahkan dengan penambahan rintangan-rintangan yang akan dihadapi oleh agen, dan menyiapkan mekanisme pengambilan gambar dari sensor untuk diberikan ke *Convolutional Neural Network* untuk menghasilkan agen yang dapat mendeteksi objek. Terkait simulator disini peneliti menggunakan *Webots* sebagai simulator saat tahapan pelatihan berbasis *Deep Imitation Learning*. Beberapa skenario simulasi disiapkan agar agen memiliki pengalaman bertemu rintangan yang bervariasi. Diharapkan variasi pengalaman tersebut menghasilkan agen yang dapat bermanuver ketika menghadapi rintangan dalam kondisi apapun, kemampuan ini dikenal juga sebagai kemampuan generalisasi. Gambar 5 menunjukkan salah satu skenario dengan 8 buah

hambatan (*obstacle*) yang di tandai oleh persegi berwarna biru untuk melatih agen dalam menghindari tabrakan (*collision avoidance*). Rute yang diambil digambarkan oleh garis berwarna merah yang mengitari jalan pada ruas kanan serta menghindari rintangan yang ada.



Gambar 5. Salah satu skenario eksperimen yang digunakan

Pada tahapan pengumpulan *dataset*, citra yang didapatkan dari sensor kamera disimpan kedalam database demonstrasi dengan *Frame Per Second* (FPS) sebesar 10 FPS. Selain itu informasi yang terkait dengan manuver yaitu sudut setir (*steer angle*) ikut direkam dan digabungkan menjadi himpunan informasi yang dikumpulkan menjadi file *.json* dengan id unik pada setiap tahapan waktu penyimpanan sesuai FPS yang digunakan.

Pada tahapan pelatihan, *dataset* yang sudah terkumpul pada database demonstrasi kemudian dilakukan pemrosesan citra dengan beberapa tahapan. Pertama adalah perubahan ukuran gambar sehingga memiliki lebar (*width*) sebesar 128 pixel, tinggi (*height*) sebesar 64 pixel, dan dimensi sebanyak 3. Dimensi ketiga ini mengacu pada kanal warna yang terdiri dari komponen merah, hijau, dan biru (RGB), yang secara bersama-sama membentuk representasi visual dari lingkungan sekitar yang ditangkap oleh kamera. Dengan pengaturan dimensi gambar seperti ini, informasi yang diperoleh dari sensor kamera dapat diakuisisi dan diolah dengan presisi, memungkinkan agen untuk membentuk persepsi yang akurat dan mendalam tentang lingkungan simulasi dalam menghindari rintangan.

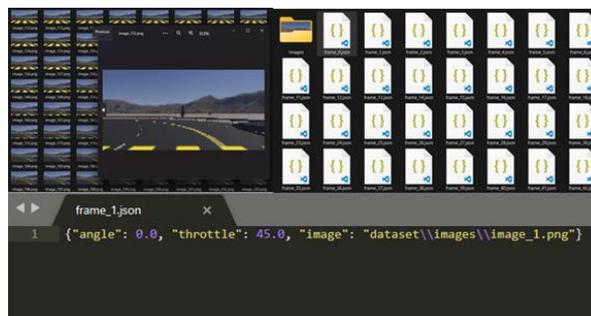
Selanjutnya dilakukan pengambilan *dataset* yang telah dikodekan dalam format *JSON*. *Dataset* akan dipecah menjadi data latih yang tersusun atas *X_train* dan *y_train* serta data uji yang tersusun atas *X_test* dan *y_test* dengan persentase masing-masing 75% dan 25%. Label arah setir yang semula berbentuk angka akan diubah menjadi representasi biner menggunakan teknik *one-hot encoding*. Kemudian, gambar-gambar yang ada dalam *dataset* akan dikonversi menjadi array numerik menggunakan array *NumPy*, dan juga diproses untuk normalisasi sesuai dengan kebutuhan model.

Selanjutnya, jika opsi pengaturan ulang *dataset* (*balanced*) diaktifkan, *dataset* akan disesuaikan agar jumlah sampel dari setiap kelas arah setir menjadi sama. Langkah ini bertujuan untuk mencegah ketidakseimbangan pada *dataset*. Setelah melakukan proses ini, *dataset* akan lebih siap digunakan untuk melatih model *Convolutional Neural Network*. Kemudian di bagian DNN yang diwakili CNN pada lapisan luaran terakhir digunakan fungsi aktivasi softmax yang terdiri dari 3 unit, sesuai dengan jumlah kelas arah setir (kiri, lurus, kanan).

Kompilasi model melibatkan pengaturan fungsi kerugian (*loss function*) dengan *categorical crossentropy* yang akan mengukur sejauh mana prediksi model cocok dengan perilaku yang direkam pada database demonstrasi. Model dilatih selama 500 epoch. Penggunaan *callback EarlyStopping* membantu mencegah overfitting dengan menghentikan pelatihan jika performa tidak membaik selama 3 epoch berturut-turut. Ini membantu model belajar secara efisien dan menghasilkan performa yang lebih baik pada data baru yang ditangkap oleh sensor kamera saat penambahan *dataset*. Evaluasi ini memberikan wawasan mengenai performa model, memungkinkan kita memahami seberapa baik model mampu menggeneralisasi pada data baru. Setelah evaluasi, model akan disimpan untuk penggunaan lebih lanjut dalam format file *.h5*. Proses penyimpanan ini memungkinkan model untuk dimuat kembali di masa depan untuk diterapkan tanpa perlu menjalani pelatihan ulang.

Penelitian ini juga menggunakan metode pelatihan dan evaluasi secara bergantian yaitu pengumpulan *dataset* sebanyak 1000 data, pelatihan, dan evaluasi. Dari hasil ujicoba agen mulai menunjukkan perilaku yang memuaskan dan generalisasi yang cukup baik ketika menyentuh 7000 *dataset* untuk menghasilkan generalisasi yang cukup baik.

Pelatihan dihentikan saat agen sudah dapat melewati semua rintangan. Proses pelatihan diulangi beberapa kali untuk memastikan arsitektur yang digunakan handal dan bukan kebetulan. Hasilnya data optimal yang dibutuhkan fluktuatif antara 8.278 hingga diatas 10.000an data hasilnya. Gambar 6 berikut menunjukkan bentuk *dataset* yang digunakan.



Gambar 6. *Dataset* yang terkumpul pada database demonstrasi

3. Hasil dan Pembahasan

Setelah melakukan proses pelatihan maka kemudian dilakukan proses evaluasi untuk menguji performa agen. Model yang sebelumnya telah melalui proses pelatihan (.h5) akan dimuat dan digunakan untuk melakukan prediksi terhadap gambar-gambar yang diambil dari kamera kendaraan. Prediksi ini kemudian akan digunakan untuk menghasilkan sudut kemudi yang sesuai dengan situasi visual yang dihadapi oleh kendaraan pada simulator webots, ilustrasi lingkungan simulasi ditunjukkan pada Gambar 7.



Gambar 7. Pengujian agen di webots

Dalam serangkaian percobaan yang dilakukan, hasilnya menunjukkan bahwa model yang telah dilatih sudah mampu untuk melewati serta menghindari semua rintangan pada lingkungan simulasi. Rintangan yang dimaksud adalah kemunculan tong besi pada yang muncul secara acak dengan jumlah acak pula. Rintangan ini disimpan pada jalur yang dilalui kendaraan untuk sampai ke tujuan. Tempat yang digunakan pada simulasi hanya satu kota, namun tempat titik awal, akhir, dan rutenya diacak yang jika digabungkan dengan kemunculan rintangan yang juga acak pada setiap simulasi maka dapat dianggap bahwa agen selalu berada pada tugas navigasi yang baru.

Agan dapat mengidentifikasi rintangan, mengambil keputusan yang tepat, dan mengatur kemudi kendaraan secara otonom sehingga menghindari tabrakan dengan rintangan yang ada. Keberhasilan model dalam menghadapi tantangan ini memberikan bukti potensi kemampuannya dalam mengemudikan kendaraan secara lebih aman dan efisien. Penelitian ini melakukan pengujian model untuk setiap 1000 *dataset* untuk mengetahui trend performa agen seiring penambahan *dataset*, sehingga dapat diketahui bahwa minimal diperlukan 8000 *dataset* menggunakan arsitektur yang diusulkan untuk agen belajar menghindari rintangan. Penelitian ini lebih baik dibandingkan dengan beberapa penelitian lain misalkan [16] yang memerlukan hingga diatas 15.000 *step/dataset* dibandingkan dengan 10.000an *step/dataset* pada penelitian ini. Tabel 1 menunjukkan detail hasil pengujian untuk penghindaran rintangan.

Tabel 1. Hasil Pengujian Penghindaran Rintangan

Jumlah Dataset	Lolos	Gagal
1 Putaran Lap 1000 Dataset	3	7
1 Putaran Lap 2000 Dataset	4	6
1 Putaran Lap 3000 Dataset	4	6
1 Putaran Lap 4000 Dataset	7	3
2 Putaran Lap 5000 Dataset	9	1
2 Putaran Lap 6000 Dataset	8	2
2 Putaran Lap 7000 Dataset	9	1
2 Putaran Lap 8000 Dataset	10	0
2 Putaran Lap 8278 Dataset	10	0

4. Kesimpulan

Pembangunan agen otonom yang dapat melakukan penghindaran rintangan ternyata dapat dilakukan secara *end-to-end* berbasis DIL, beberapa teknik pemrosesan terbukti dapat mengoptimalkan pelatihan dan meringankan beban komputasi. Pemilihan luaran kontrol mana yang dilatih dan mana yang dikodekan secara manual dapat menghasilkan agen otonom yang lebih cepat. Pada penghindaran rintangan ini terdapat dua kontrol yang dominan yaitu *gas/throttle* dan *setir/angle*, *throttle* dikodekan secara manual dengan nilai konstan di 50% yang menghasilkan kecepatan di 30 km per jam sedangkan *angle* dilatih menggunakan *Deep Imitation Learning*. Dengan melatih fitur lebih sedikit maka agen otonom dapat memiliki kemampuan yang diharapkan dengan lebih cepat dan dataset yang relatif lebih sedikit. Tentu saja pendekatan seperti ini memiliki kekurangan jika kemampuan yang ingin dilatih lebih dari satu atau dikenal dengan *Multi task training* yang akan memerlukan pendekatan lain yang lebih cocok misalkan *Curriculum Based* [17] atau *Continual Learning* [18]

Ada beberapa isu dan prospek yang menarik untuk dilakukan penelitian lebih lanjut, seperti bagaimana jika penghindaran rintangan melibatkan hambatan yang benar-benar menutupi jalan, bagaimana jika agen juga perlu untuk mematuhi rambu, dan bagaimana menggabungkan agen otonom dengan mode yang bisa diatur seperti kecepatan minimal atau maksimal yang bisa diatur oleh manusia namun membiarkan sisi lain dari agen otonom tetap otomatis, aspek kombinasi antara sifat terkontrol dan otonom seperti ini menarik untuk diangkat.

Ucapan Terimakasih

Terimakasih kepada Fakultas Teknik Universitas Pasundan yang telah mendukung penelitian ini secara moril dan material.

Daftar Rujukan

- [1] T. de Swarte, O. Boufous, and P. Escalle, "Artificial intelligence, ethics and human values: the cases of military drones and companion robots," *Artif Life Robot*, vol. 24, no. 3, pp. 291–296, Sep. 2019, doi: 10.1007/s10015-019-00525-1.

- [2] C. Wu *et al.*, "UAV autonomous target search based on deep reinforcement learning in complex disaster scene," *IEEE Access*, vol. 7, pp. 117227–117245, 2019, doi: 10.1109/ACCESS.2019.2933002.
- [3] J. Dai, R. Li, Z. Liu, and S. Lin, "Impacts of the introduction of autonomous taxi on travel behaviors of the experienced user: Evidence from a one-year paid taxi service in Guangzhou, China," *Transp Res Part C Emerg Technol*, vol. 130, Sep. 2021, doi: 10.1016/j.trc.2021.103311.
- [4] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, "Automatic obstacle avoidance of quadrotor UAV via CNN-based learning," *Neurocomputing*, vol. 402, pp. 346–358, Aug. 2020, doi: 10.1016/j.neucom.2020.04.020.
- [5] S. Back, G. Cho, J. Oh, X. T. Tran, and H. Oh, "Autonomous UAV Trail Navigation with Obstacle Avoidance Using Deep Neural Networks," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 100, no. 3–4, pp. 1195–1211, Dec. 2020, doi: 10.1007/s10846-020-01254-5.
- [6] F. Ye, S. Zhang, P. Wang, and C. Y. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 1073–1080. doi: 10.1109/IV48863.2021.9575880.
- [7] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5068–5078, Jun. 2022, doi: 10.1109/TITS.2020.3046646.
- [8] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A Survey of End-to-End Driving: Architectures and Training Methods," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 4, pp. 1364–1384, Apr. 2022, doi: 10.1109/TNNLS.2020.3043505.
- [9] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluca, and L. M. Bergasa, "Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator," *Sensors*, vol. 22, no. 21, Nov. 2022, doi: 10.3390/s22218373.
- [10] B. Osiński *et al.*, "CARLA Real Traffic Scenarios -- novel training ground and benchmark for autonomous driving," Dec. 2020, [Online]. Available: <http://arxiv.org/abs/2012.11329>
- [11] X. Liang, Y. Liu, T. Chen, M. Liu, and Q. Yang, "Federated Transfer Reinforcement Learning for Autonomous Driving," Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.06001>
- [12] A. Kusari *et al.*, "Enhancing SUMO simulator for simulation based testing and validation of autonomous vehicles," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 829–835. doi: 10.1109/IV51971.2022.9827241.
- [13] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluca, and L. M. Bergasa, "Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator," *Sensors*, vol. 22, no. 21, Nov. 2022, doi: 10.3390/s22218373.
- [14] M. Franchi, "Webots.HPC: A Parallel Robotics Simulation Pipeline for Autonomous Vehicles on High Performance Computing," Aug. 2021, [Online]. Available: <http://arxiv.org/abs/2108.00485>
- [15] D. H. Lee, K. L. Chen, K. H. Liou, C. L. Liu, and J. L. Liu, "Deep learning and control algorithms of direct perception for autonomous driving," *Applied Intelligence*, vol. 51, no. 1, pp. 237–247, Jan. 2021, doi: 10.1007/s10489-020-01827-9.
- [16] E. Zhang, H. Zhou, Y. Ding, J. Zhao, and C. Ye, "Learning how to avoiding obstacles for end-to-end driving with conditional imitation learning," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Nov. 2019, pp. 108–113. doi: 10.1145/3372806.3372808.
- [17] L. Anzalone, P. Barra, S. Barra, A. Castiglione, and M. Nappi, "An End-to-End Curriculum Learning Approach for Autonomous Driving Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19817–19826, Oct. 2022, doi: 10.1109/TITS.2022.3160673.
- [18] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing Change: Continual Learning in Deep Neural Networks," *Trends in Cognitive Sciences*, vol. 24, no. 12. Elsevier Ltd, pp. 1028–1040, Dec. 01, 2020. doi: 10.1016/j.tics.2020.09.004.