



Perancangan *Data Pipeline Computer Vision* untuk Mendukung Pelatihan Agen Otonom dalam Mengenali Rambu Lalu Lintas

Mellia Liyanthy¹

¹Fakultas Teknik, Program Studi Teknik Informatika, Universitas Pasundan, Bandung, Indonesia

Email: 1liyanthy@unpas.ac.id

Abstrak

Kendaraan otonom merupakan alat transportasi yang dibangun dari sensor yang dipasang dengan perangkat komputasi tertentu, kemudian diproses oleh algoritma yang akan mengeluarkan kontrol untuk menggantikan manusia dalam mengemudi kendaraan secara otomatis. *Data pipeline computer vision* merupakan salah satu hal yang diperlukan untuk mendukung agen cerdas dalam mengenali rambu-rambu lalu lintas, sehingga agen mampu memproses dan menganalisa data *image* yang mampu membantu pengemudi menghindari sejumlah besar potensi bahaya dan meningkatkan pengalaman berkendara. Tahapan perancangan *Traffic Sign Recognition* (TSR) untuk mendukung pelatihan agen otonom dalam mengenali rambu lalu lintas dimulai dengan membuat *Dataset* Rambu Lalu Lintas Indonesia (DSRLI), *pre-processing*, kemudian melakukan *Traffic Sign Detection* (TSD) dan TSR dan akhirnya melakukan aksi berdasarkan informasi yang dikumpulkan. Hasil dari penelitian yang telah dilakukan berupa pendekatan baru untuk mempercepat proses deteksi rambu lalu lintas yang diusulkan berupa *complete data pipeline* untuk TSR, termasuk deteksi dan pengenalan tanda yang dijelaskan, serta data set baru yang spesifik untuk lalu lintas di Indonesia.

Keywords: Kendaraan otonom, *Pipeline Computer Vision*, *Traffic-Sign Recognition*

1. PENDAHULUAN

Beberapa tahun terakhir, banyak metode untuk *Traffic Sign Detection* (TSD) dan *Traffic Sign Recognition* (TSR) berdasarkan teknik *Computer Vision* yang telah diusulkan. Karena rambu jalan memiliki warna standar, beberapa pendekatan didasarkan pada segmentasi warna untuk menemukan rambu pada gambar input [1]. Pendekatan *sliding windows* yang diusulkan, sebagian besar mengandalkan penggunaan *Histogram of Oriented Gradients* (HOG), *Integral Channel Features* (ICF) dan



modifikasinya untuk mengekstrak fitur visual diskriminatif dari gambar. Metode ini bertujuan menggabungkan kecepatan teknik segmentasi warna dengan keakuratan pendekatan sliding windows, khususnya HOG [2].

Data pipeline computer vision merupakan *data flows* yang dimulai dengan pengumpulan data hingga penyimpanannya, kemudian data tersebut akan digunakan untuk melatih agen cerdas dalam konsep *Artificial Intelligent* (AI) dan menimplementasikannya [3]. Idealnya *pipeline computer vision* merupakan pengaturan teknis yang menghubungkan penyimpanan data ke berbagai data preparation dan peralatnya, kemudian terhubung melalui API dari model pembelajaran agen dari produk yang akan dibangun [4].

Usaha untuk mencapai kendaraan *Ego Vehicle* (EV) yang mampu untuk mengambil keputusan tepat terkait aman dan nyaman tersebut bukanlah masalah yang sederhana. Skenario pada lalu lintas urban sangat kompleks disebabkan oleh perilaku objek lalu lintas lain yang dinamis. *Data pipeline computer vision* dapat digunakan untuk mendukung tahap pembelajaran EV untuk mengenali rambu sehingga dapat bertindak sesuai dengan aturan lalu lintas yang berlaku [5-6].

Berdasarkan penjelasan di atas maka penelitian ini bertujuan untuk membuat perancangan *Data pipeline computer vision* untuk membantu pelatihan agen otonom untuk dapat mematuhi rambu-rambu lalu lintas. Kontribusi yang diharapkan dari penelitian ini adalah (i) Sebuah pendekatan baru untuk mempercepat proses deteksi rambu lalu lintas yang diusulkan. (ii) Sebuah *complete pipeline*, termasuk deteksi dan pengenalan tanda yang dijelaskan. (iii) Data set baru yang spesifik untuk lalu lintas di Indonesia.

2. METODOLOGI

Metode yang digunakan pada penelitian perancangan *Data pipeline computer vision* untuk membantu pelatihan agen otonom diawali dengan pengumpulan data rambu-rambu lalu lintas untuk menghasilkan Dataset yang akan digunakan oleh agen otonom tersebut, kemudian dilakukan analisis data untuk mengelompokkan data *training* dan *testing* yang akan digunakan baik pada *subset* deteksi maupun subset klasifikasi, tahap terakhir melakukan perancangan *data pipeline computer vision* untuk pengenalan rambu lalu lintas tersebut.

2.1. Pengumpulan Data

Pengumpulan data dilakukan untuk membuat *Dataset* Rambu Lalu Lintas Indonesia (DSRLI) yang akan digunakan sebagai data *training* dan *testing*. Pengumpulan data dilakukan dengan melakukan pencarian gambar rambu-rambu lalu lintas dari berbagai sumber *website*, kemudian melengkapi rambu-rambu lalu lintas yang belum ada dengan cara mengcapture rambu-rambu lalu lintas yang ada di jalan raya yang dihasilkan sebagian dari video HD (ukuran bingkai 1280×720 yang direkam pada 10 fps) yang diambil menggunakan kamera dan berbagai *smartphone*. Seperti yang disarankan oleh penelitian terdahulu, pada penelitian ini *Dataset* yang dibuat sudah meningkatkan keragamannya dengan menambahkan gambar yang diambil pada malam hari dan saat ada kabut (lihat Gambar 1), sehingga dapat meningkatkan keberadaan *false positive*.



Gambar 1. Contoh Gambar yang Ada pada DSRLI

2.2. Analisis Data

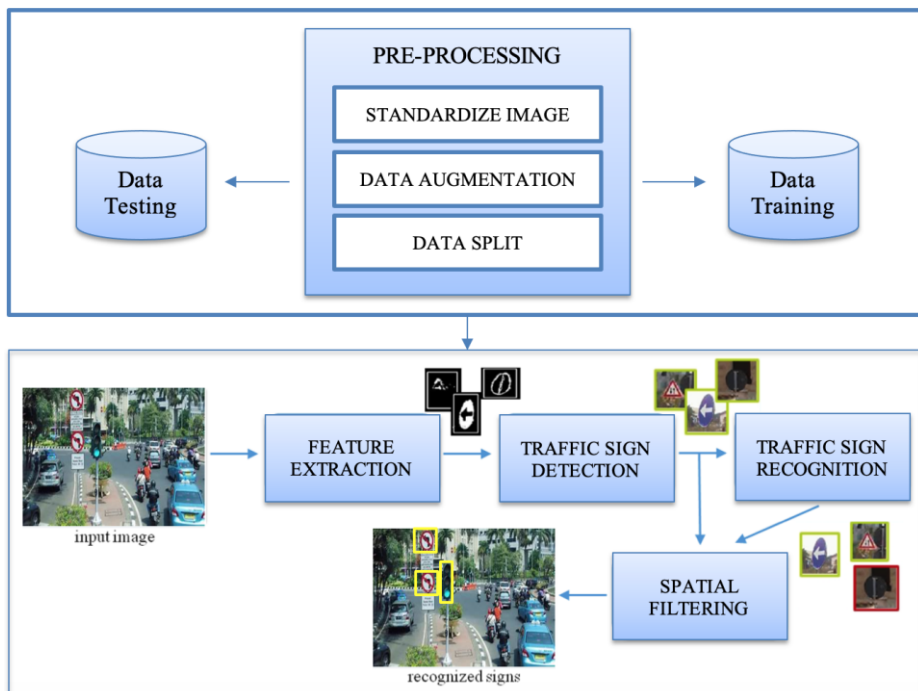
DSRLI yang dibuat, awalnya dihasilkan dari 43.289 gambar yang diambil dari lebih dari 14 jam video yang direkam di berbagai tempat di Bandung. Gambar kemudian diambil sampelnya untuk memenuhi persyaratan. Semua gambar dalam kumpulan data dikelompokkan ke dalam kelas. Secara khusus, gambar pelatihan untuk deteksi dan klasifikasi diatur dalam folder, di mana setiap folder mewakili superclass atau sub-class tunggal. Gambar beranotasi tersedia untuk pengujian dan file teks dengan informasi dan anotasi yang lebih akurat.

2.3. Perancangan

Perancangan *Data pipeline computer vision* dilakukan dengan mengkombinasikan langkah-langkah perancangan *data pipeline* pada pendekatan *Computer Vision* dengan pendekatan pada *Machine Learning*. Pertama melakukan pre-processing terhadap data input berupa image atau video, kemudian dilakukan feature extraction untuk mendefinisikan sebuah objek berupa informasi bentuk dan warna sehingga dapat mengidentifikasi rambu lalu lintas berdasarkan informasi tersebut, selanjutnya melakukan *Traffic Sign Detection* (TSD) dan langkah terakhir melakukan *Traffic Sign Recognition* (TSR) untuk memprediksi class dari rambu lalu lintas tersebut dengan menggunakan algoritma yang dihasilkan untuk pelatihan agen cerdasnya [7-9].

3. HASIL DAN PEMBAHASAN

Hasil perancangan *Data Pipeline Computer Vision* yang diusulkan ditunjukkan pada Gambar 2.



Gambar 2. *Data pipeline computer vision* yang diusulkan

Pada awalnya, dilakukan tahap *pre-processing* untuk membuat Dataset yang terbagi menjadi data *training* dan *testing*. Segmen dari gambar masukan yang memiliki atribut warna yang mirip dengan model rambu jalan yang dipilih dan membuat rangkaian awal Regions of Interest (ROI). Kemudian, fitur diekstraksi dari setiap ROI dan direpresentasikan sebagai deskriptor *Histogram of Oriented Gradient* (HOG). Setelah itu, proses deteksi *multi-scale*, berdasarkan model *Support Vector Machine* (SVM) yang sudah dilatih. Informasi tentang posisi dan ukuran setiap kemungkinan tanda disimpan untuk digunakan pada tahap selanjutnya. Pada saat yang sama, setiap deteksi diteruskan ke modul pengenalan berbasis *Convolutional Neural Network* (CNN).

3.1. Pre-Processing

Tahapan *pre-processing* meliputi:

- 1) *Standardize images* untuk mengatasi salah satu kendala penting yang ada di beberapa algoritma pembelajaran mesin, seperti CNN, yaitu kebutuhan untuk mengubah ukuran gambar dalam kumpulan data ke dimensi terpadu. Gambar harus diproses sebelumnya dan diskalakan agar memiliki lebar dan tinggi yang identik sebelum dimasukkan ke algoritma pembelajaran. Semua gambar sampel ukurannya distandarkan menjadi 28x28.
- 2) *Data augmentation* merupakan teknik *pre-processing* umum lainnya yang melibatkan penambahan kumpulan data yang ada dengan versi gambar yang masih mengandung *noise*. Penskalaan, rotasi, dan transformasi *affine* lainnya dilakukan untuk memperbesar kumpulan data dan mengekspos ke berbagai variasi gambar. Ini membuatnya lebih mungkin bahwa model tersebut mengenali objek ketika mereka muncul dalam bentuk apa pun.
- 3) Data Split merupakan tahap akhir pada *pre-processing* untuk membagi dataset menjadi data *training* dan *testing*. DSRLI yang dibuat terdiri dari 1.416 gambar untuk pelatihan dan 471 untuk pengujian untuk subset deteksi, sedangkan subset klasifikasi berisi 8.048 sampel untuk pelatihan dan 1.206 untuk pengujian.

3.2. Feature Extraction

Feature Extraction merupakan tahap untuk pengambilan ciri/*feature* dari suatu objek yang mencirikan rambu lalu lintas tertentu, yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Feature yang digunakan meliputi warna dan bentuk dari rambu lalu lintas tersebut.

3.3. Traffic Sign Detection (TSD)

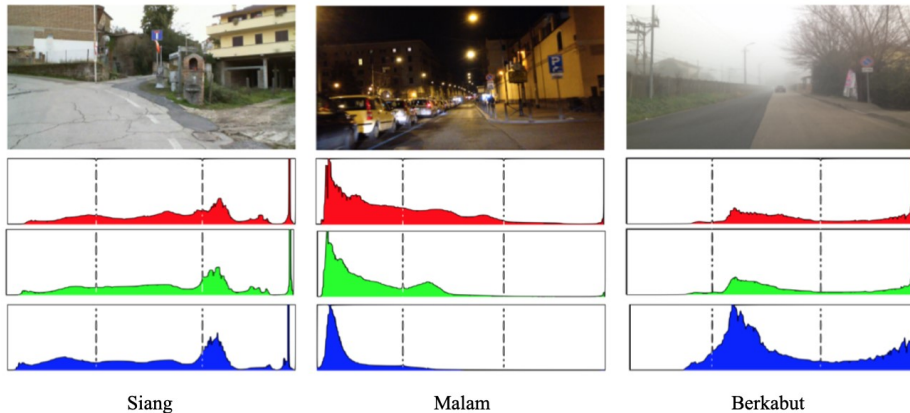
TSD yang dirancang menggunakan kombinasi metode berbasis warna dan bentuk untuk mendeteksi rambu lalu lintas. Metode berbasis bentuk mencakup operasi normalisasi selain pemrosesan gambar multi-scale, untuk menangani kompleksitas yang disajikan dalam gambar jalanan pada umumnya. Karena piksel dari rambu lalu lintas biasanya mewakili persentase kecil dari semua piksel dalam input image, untuk mengurangi ruang pencarian awal. Kemudian diasumsikan bahwa rambu lalu lintas memiliki saturasi dan warna cerah yang berbeda dengan lingkungan sekitar, agar mudah dikenali oleh pengemudi. Asumsi ini membantu dalam menentukan cakupan gambar dengan menggunakan detection windows yang hanya melingkupi sebagian kecil dari patch yang diekstraksi dari gambar aslinya.

Ekstraksi ROI dilakukan dengan memanfaatkan ruang warna Improved Hue, Saturation, dan Luminance (IHSL), yang berguna untuk memperhitungkan perubahan pencahayaan karena komponen kromatik dan akromatik tidak bergantung pada IHSL. Gambar asli diubah dari RGB ke IHSL dengan menerapkan operasi thresholding yang ada pada Persamaan (1).

$$IHSL(x, y) = \begin{cases} 0, & \text{if } \text{Max}(I_R(x, y), I_G(x, y), I_B(x, y)) = I_G(x, y) \vee \\ & (|I_R(x, y) - I_G(x, y)| < \zeta \wedge |I_B(x, y) - I_G(x, y)| < \gamma) \\ F_{ihsl}(I_R(x, y), I_G(x, y), I_B(x, y)), & \text{otherwise} \end{cases} \quad (1)$$

di mana $IHSL(x, y)$ adalah nilai color space IHSL untuk piksel pada posisi (x, y) , yang memiliki nilai I_R , I_G , dan I_B dari RGB. Fungsi $F_{ihsl}(\cdot)$ merepresentasikan konversi biasa $RGB_{\text{colorspace}} \rightarrow IHSL_{\text{colorspace}}$. Perlu diketahui bahwa piksel RGB dengan persentase hijau yang cukup besar (misalnya, mewakili vegetasi) dibuang. Ambang batas ζ dan γ memainkan peran penting dalam memfilter piksel lebih lanjut (misalnya milik jalan

dan langit), sehingga mengurangi waktu komputasi untuk konversi IHSL dan langkah deteksi berikutnya.



Gambar 3. Histogram dari Tiga Kondisi Jalan yang Berbeda

Nilai untuk ζ dan γ diatur berdasarkan kondisi pencahayaan dalam gambar, distribusi warna yang diwakili oleh histogram RGB digunakan untuk mendeteksi kondisi pencahayaan (lihat Gambar 3). Setiap histogram RGB dibagi menjadi tiga area yang sama. Pembagian ini membantu menentukan di mana informasi warna diringkas dan menemukan nilai puncak di setiap histogramnya. Pada kondisi siang hari, sebaran informasi warna terletak di bagian kanan histogram, dengan nilai puncak di sisi paling kanan. Kondisi malam hari memiliki jumlah piksel yang lebih tinggi, yang terletak di area kiri, sedangkan dalam kondisi berkabut jumlah piksel yang lebih besar tersebar di area tengah dan kanan. Secara lebih formal, distribusi warna pada setiap area histogram dihitung menggunakan Persamaan (2).

$$\epsilon_n = \frac{\sum_n H(i), \text{ if } H(i) \geq \alpha}{\sum_{\forall_n} H(i)} \quad (2)$$

di mana ϵ_n mewakili distribusi setiap saluran warna RGB pada salah satu dari tiga area ($n \in \{1,2,3\}$), $H(i)$ adalah jumlah piksel dengan nilai warna $i \in [0, 255]$, dan α diatur ke $0,3 * \max(H(i))$, $\forall_i, \epsilon_n$ maksimum digunakan untuk mendeteksi kondisi pencahayaan pemandangan dan menyesuaikan nilai γ dan ζ .

Citra biner akhir dihitung dengan menerapkan pendekatan (1), di mana metode *Normalized Hue-Saturation* (NHS) dan langkah-langkah pasca-

pemrosesan digunakan untuk menemukan piksel potensial milik rambu-rambu jalan. Operator morfologi erosi dan dilation diterapkan pada citra biner untuk penghilangan noise. Piksel yang tersisa dikelompokkan ke dalam kontur, yang kemudian difilter menurut ukurannya. Kotak pembatas dari setiap kontur yang diekstraksi dihitung dan dikelompokkan ke dalam kelas spasial sesuai dengan jarak dan setiap kotak pembatas yang tumpang tindih. Prosedur ini berguna untuk menangani rambu jalan yang partially occluded, dengan banyak warna, atau rusak. Gambar 4 menunjukkan segmentasi warna bersama dengan prosedur ekstraksi kontur yang mengarah ke ekstraksi patch (disorot pada baris terakhir dari gambar).

Gambar asli ada di baris pertama. Baris kedua menunjukkan hasil RGB *thresholding*, sedangkan baris ketiga berisi ekstraksi kontur. Di kolom pertama nilai digunakan $\zeta = 15$ dan $\gamma = 25$, di kolom kedua $\zeta = 40$ dan $\gamma = 25$ dan di kolom ketiga $\zeta = 15$ dan $\gamma = 20$.



Gambar 3. Segmentasi Warna dan Ekstraksi Kontur

Deskriptor visual kemudian dihitung di atas patch yang diekstraksi. Untuk HOG, menggunakan jendela deteksi 40×40 dengan ukuran blok 10×10 dan langkah 2×2 . Tahap training dilakukan sebanyak tiga putaran secara total, menambahkan hard negative pada training images di setiap putaran. Untuk putaran pertama, menggunakan sampel positif DSRLI bersama dengan custom negatives, ditambahkan sekitar 2000 sampel. Pada putaran kedua, prosedur yang sama diterapkan pada data set rambu lalu lintas yang dibuat. Positives samples dari data set rambu lalu lintas yang dibuat, bersama dengan semua hard negative yang dikumpulkan dari langkah sebelumnya digunakan untuk putaran terakhir dari tahap training. Tahap segmentasi warna membantu mengurangi deteksi false positive dan mempercepat keseluruhan prosedurnya.

3.4. Traffic Sign Recognition (TSR)

Modul recognition yang dirancang, didasarkan pada CNN dan dibangun di atas kerangka TensorFlow Deep Learning. Semua gambar diambil sampelnya menjadi 28×28 . Untuk data set rambu lalu lintas yang dibuat, sampel jitter (augmented) juga digunakan. Gambar diputar, diskalakan, dan diterjemahkan secara acak. Nilai maksimum untuk gangguan adalah $[-15, 15]$ untuk rotasi, $[-4, 4]$ untuk translasi dan faktor $[0.8, 1.2]$ untuk penskalaan. Kumpulan data yang diperbesar empat kali lebih besar dari aslinya.

Pendekatan Deep Learning memiliki dua arsitektur yang berbeda, Single-Scale Convolutional Neural Network dan versi modifikasi dari arsitektur multi-scale yang diusulkan untuk menandingi modul TSR dari penelitian terdahulu. Arsitektur Single-Scale menghadirkan dua lapisan konvolusional, masing-masing diikuti oleh pooling layer, dan dua sequential fully connected layer dengan rektifikasi. Layer terakhir adalah pengklasifikasi linier softmax. CNN multi-scale dan arsitektur skala tunggal yang dijelaskan di atas. Model ini menyajikan dua tahap convolutional layers, dua local fully connected layer dengan aktivasi linier yang diperbaiki dan pengklasifikasi softmax. Adapun kasus single scale, setiap convolutional layers menyajikan pooling layer. Keluaran dari tahap konvolusional pertama memiliki dua percabangan, pertama feeds tahap berikutnya dengan menggunakan metodr feed-forward, cabang yang kedua berfungsi sebagai bagian dari masukan untuk lapisan pertama yang terhubung sepenuhnya. Input dari layer tersebut dibangun dari output tahap pertama dan output tahap kedua.

3.5. Spatial Filtering

Modul terakhir dari data *pipeline computer vision* yang diusulkan, berfungsi untuk menggabungkan informasi dari modul deteksi dan modul klasifikasi. Algoritma 1 berisi detail tentang langkah dari modul terakhir ini. Nilai probabilitas yang diberikan ke masing-masing patch yang tidak dikenali sebagai rambu lalu lintas tertentu, dengan mempertimbangkan posisinya dan patch tetangganya.

Algorithm 1: Spatial Neighbourhood Filtering

Input : unrecognised detections α , recognized detections β .
Output: Accepted detection

```

1  for  $\forall d: d \in \beta$  do
2    1:  $M \leftarrow \text{maximum range for } d$ ;
3    2: for  $\forall c: c \in \alpha \wedge c \text{ in range } M$  do
4      i:  $p_d \leftarrow \text{compareSize}(d, c)$ ;
5      ii:  $p_d \leftarrow \text{compareDistance}(d, c)$ ;
6    end
7    3: if  $p_d > \text{threshold}$  then
8      accept  $d$ ;
9    else
10     discard  $d$ ;
11   end
12 end
```

Proses ambang diterapkan lebih lanjut pada probabilitas akhir untuk menyaring patch yang tidak dipercaya dari daftar akhir. Perlu dicatat bahwa dalam aplikasi nyata, seperti driver assistance systems, pengemudi tidak boleh diperingatkan karena deteksi palsu. Memang, meskipun solusi yang diusulkan membantu meningkatkan akurasi sebesar 1-2% tergantung pada kumpulan data, solusi ini sangat bergantung pada subsistem pengenalan, menambahkan kemungkinan kecil untuk membuang deteksi yang benar.

Output dari modul klasifikasi dibagi menjadi deteksi terklasifikasi (yaitu, deteksi yang berhasil ditetapkan ke kelas oleh CNN) dan yang tidak terklasifikasi. Untuk setiap elemen dalam set deteksi yang tidak terklasifikasi, maka penelitian yang dilakukan mendefinisikan rentang pencarian di sekelilingnya dan menghitung, untuk setiap sampel dalam set yang terklasifikasi yang terletak di dalam rentang tersebut, dua metrik berikut: (i) kemiripan antara ukuran elemen yang tidak terklasifikasi dan yang diklasifikasikan kemudian (ii) jarak antara pusat dari dua elemen. Jika jumlah dari dua metrik perbandingan yang tercantum di atas berada

di bawah ambang batas yang diberikan, maka deteksi yang tidak terklasifikasi akan dibuang secara definitif, jika tidak maka diterima.

4. KESIMPULAN

Dalam penelitian ini pendekatan yang diusulkan menggabungkan berbagai metode canggih untuk mendapatkan hasil kompetitif pada deteksi dan klasifikasi rambu lalu lintas. Perhatian khusus telah diberikan pada fase deteksi, di mana segmentasi warna digunakan untuk mengurangi porsi gambar yang akan diproses, sehingga mengurangi waktu komputasi. Arsitektur CNN memungkinkan untuk membuang positif palsu lebih lanjut dengan menggabungkan keluaran dari modul deteksi dan pengenalan. Data Rambu Lalu Lintas Indonesia menampilkan beberapa inovasi dari kumpulan data yang ada, yaitu kondisi malam hari dan pemandangan perkotaan yang kompleks.

REFERENSI

- [1] D. Ponce and D. Forsyth, "Computer Vision: A Modern Approach," *2nd ed., Pearson*, 2011.
- [2] B. J. Chelliah, V. Chauhan, S. Mishra, and V. Sharma, "Advancement of Driverless Cars and Heavy Vehicles using Artificial Intelligence (Object Detection)," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 6183–6186, 2019.
- [3] R. Szeliski, "Computer Vision: Algorithm dan Application," *Springer*, 2011.
- [4] N. Agarwal, C.-W. Chiang, and A. Sharma, "A Study on Computer Vision Techniques for Self-driving Cars," in *Int. Conf. on Frontier Computing*, Springer Link, pp. 629–634, 2018.
- [5] M. P. Philipsen, M. B. Jensen, A. Møgelmoose, T. Moeslund, and M. Trivedi, "Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset," in *Proc. IEEE 18th Int. Conf. Intelligent Transportation Systems*, 2015.
- [6] M. B. Jensen, M. P. Philipsen, A. Møgelmoose, T. Moeslund, and M. Trivedi, "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, 2016.
- [7] J. Janai, F. Guney, A. Behl, and A. Geiger, "Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art," *Computer Science*, pp. 35–43, 2017.

- [8] D. Vitas, M. Tomic, and M. Burul, "Traffic Light Detection in Autonomous Driving Systems," *IEEE Consumer Electronics Magazine*, vol. 9, no. 4, pp. 90–96, 2020.
- [9] H. Marina, I. Soto, J. Valerio, R. Zamorano-Illanes, E. Toledo-Mercado, and R. Wang, "Automatic Traffic Light Detection Using AI for VLC," in *Proc. 13th Int. Symp. on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 446–451, 2022.