Perbandingan Kinerja YOLOv10 dan *EfficientDet* dalam Deteksi Mata Tertutup dan Mulut Terbuka sebagai Parameter Deteksi Pengemudi Mengantuk berbasis Citra Digital

Muhammad Fida Raditya[1], Andika Setiawan[2], Ilham Firman Ashari[3]
Teknik Informatika
Institut Teknologi Sumatera
Kab. Lampung Selatan, Indonesia
e-mail: [1]radityamuhammadf@gmail.com, [2] andika.setiawan@if.itera.ac.id, [3] firman.ashari@if.itera.ac.id

Correspondence : *e-mail*: radityamuhammadf@gmail.com

Diajukan: 30 Juli 2024; Direvisi: 09 Agustus 2024; Diterima: 18 Agustus 2024

Abstrak

Penelitian ini mengembangkan dan membandingkan dua framework model deteksi objek (YOLOv10 dan EfficientDet) pada gejala pengemudi mengantuk yang dapat dikenali dari aspek visual seperti mata tertutup dan mulut terbuka, serta menganalisis performa kedua model dari segi kecepatan inference dan keakuratan deteksi berdasarkan pengujian confusion matrix. Kedua model deteksi yang telah dikembangkan dengan dataset dan konfigurasi yang sama diujikan pada 1441 buah gambar, nilai yang dievaluasi berupa metrik accuracy, precision, dan recall, serta rerata kecepatan inference model deteksi saat mengolah citra yang diterima. Hasil pengujian yang dilakukan menunjukkan perbedaan signifikan pada kecepatan inference YOLOv10 yang dua kali lebih cepat dibandingkan EfficientDet dengan YOLOv10 sebesar 27,59 ms dan EfficientDet sebesar 54,12 ms. Adapun pada evaluasi keakuratan, YOLOv10 unggul dengan selisih 2% hingga 2,6% pada hasil perhitungan accuracy, precision, dan recall.

Kata kunci: *Deteksi, Mengantuk, YOLOv10, EfficientDet.*

Abstract

This research develops and compares two object detection model frameworks (YOLOv10 and EfficientDet) on drowsy driver symptoms that can be recognized from visual aspects such as closed eyes and open mouth, and analyzes the performance of both models in terms of inference speed and detection accuracy based on confusion matrix testing. The two detection models that have been developed with the same dataset and training configuration are tested on 1441 images, the evaluated metrics are accuracy, precision, and recall values, as well as the average inference speed of the detection model when processing the each image on iteration. The test results show a significant difference in terms of inference speed of YOLOv10, which is twice as fast as EfficientDet with YOLOv10 at 27.59 ms and EfficientDet at 54.12 ms. As for the accuracy evaluation, YOLOv10 excels with a difference of 2% to 2,6% on accuracy, precision and recall metrics.

Keywords*: Drowsiness, Detection, YOLOv10, EfficientDet.*

1. Pendahuluan

Data Pusiknas Bareskrim Polri Semester I Tahun 2022 mengungkapkan terdapat 12.288 kecelakaan roda empat yang diakibatkan oleh kelalaian pengemudi [1]. Kondisi mengantuk merupakan salah satu dari kelalaian pengemudi yang terjadi dalam waktu singkat namun memiliki resiko fatal, umumnya diakibatkan oleh faktor siklus tidur, obat-obatan, hingga tekanan emosional yang dialami pengemudi [2]. Berbagai pengembangan teknologi dilakukan untuk mengatasi permasalahan tersebut, salah satunya dalam implementasi *computer vision* dalam pengembangan teknologi keselamatan berkendara terintegrasi seperti *Advanced Driving Assistance System* (ADAS) [3].

Dalam identifikasi pengemudi mengantuk berbasis citra digital, kondisi mata tertutup dan mulut terbuka umum dijadikan sebagai parameter utama deteksi kondisi mengantuk seperti penelitian yang dilakukan oleh May Thu Soe, dkk. pada 2022 [4], Riyadh Ayachi, dkk. pada tahun 2023 [5], dan Petchara

Intanon pada tahun 2021 [6] yang menggunakan *facial landmarks* mata dan mulut sebagai parameter utama pendeteksian kondisi mengantuk. Untuk bisa mendapatkan kemampuan deteksi model yang baik dalam pendeteksian kondisi mengantuk yang dialami pengemudi, *dataset* Driver Monitoring Dataset (DMD) dan Yawning Detection Dataset (YawDD) dikembangkan untuk memberikan *dataset* yang dapat merepresentasikan kondisi mengantuk yang dialami pengemudi melalui serangkaian video skenario pada dataset tersebut [7], [8].

Pada pengembangan *computer vision*, berbagai *framework* dikembangkan untuk mencapai tujuan yang bervariasi, salah satunya *framework one-stage object detectors* yang pengembangannya dioptimalkan untuk mendeteksi objek secara *real-time* secara optimal dari segi kecepatan dan akurasi deteksi untuk bisa di-integrasikan dengan perangkat *edge computer* [9]. Pengembangan terbaru *framework one-stage object detector* adalah perilisan Ultralytics YOLOv10 oleh oleh Ao Wang, dkk. pada 2024 yang melakukan penyeimbangan akurasi dan kecepatan deteksi pada *detection head framework* tersebut dengan *consistent matching metric* sebagai alternatif dari *non-maximum supression* (NMS) yang akan menurunkan *latency* saat *inference* tanpa mengurangi akurasi deteksi [10]. Dikembangkan juga EfficientDet pada 2020 oleh Minxing Tan, dkk. yang merupakan salah satu bagian dari TensorFlow 2 Detection Model Zoo, *framework* ini berfokus pada skalabilitas kompleksitas model yang dapat disesuaikan kebutuhannya dengan *resource* yang tersedia [11].

Dua buah *framework* tersebut telah diimplementasikan dalam pengembangan deteksi pengemudi mengantuk, seperti yang dilakukan oleh May Thu Soe, dkk. pada 2022 yang mengembangkan model deteksi distraksi yang dialami pengemudi menggunakan YOLOv2 dan berhasil mendapatkan akurasi deteksi sebesar 95,9%, penelitian ini tidak memberikan gambaran mengenai kecepatan deteksi yang didapatkan melalui metrik apapun. Riadh Ayachi, dkk. pada tahun 2023 mengembangkan model deteksi pengemudi mengantuk menggunakan EfficientDet-D0 yang berhasil mendapatkan rata-rata akurasi 96% dan kecepatan deteksi dengan perhitungan *frame per second* (FPS) sebesar 43 FPS pada perangkat laptop dengan *graphical processing unit* (GPU) GTX 960.

Penelitian ini dilakukan untuk bisa mendapatkan gambaran mengenai perbandingan performa model deteksi yang dari *kedua framework* terbaru yang pada penelitian sebelumnya telah diimplementasikan pada pengembangan sistem deteksi pengemudi mengantuk, aspek yang diamati yakni berupa tingkat keakuratan deteksi model yang dievaluasi dengan perhitungan *accuracy, precision, dan recall* berdasarkan hasil pengujian *confusion matrix* terhadap proporsi dataset yang digunakan, serta perhitungan kecepatan *inference* model deteksi dengan perekaman waktu *inference* yang dibutuhkan saat proses deteksi dilakukan. Evaluasi keakuratan dilakukan untuk bisa memberi gambaran pada kemampuan model melakukan deteksi dan resiliensi terhadap kondisi *false positive*, dan evaluasi kecepatan *inference* dilakukan untuk dapat memberi gambaran awal mengenai kecepatan model yang dihasilkan saat digunakan dalam melakukan deteksi pengemudi mengantuk sebagai bahan pertimbangan saat akan diimplementasikan pada perangkat *single board computer* (SBC) atau *system on chip* (SoC) ADAS untuk pengembangan lebih lanjut.

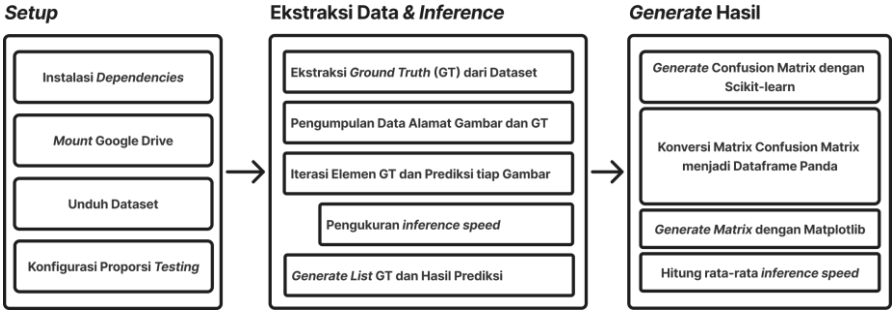
2. Metode Penelitian

Penelitian ini menggunakan total 7780 buah *frame* yang merepresentasikan kondisi normal, terpejam, berkedip, dan menguap yang dialami pengemudi untuk diklasifikasikan menjadi dua kelas, yakni *closed eyes* dan *yawn. Dataset* didapatkan dari proses *slicing* video yang didapatkan dari DMD dan YawDD dan dianotasi secara manual pada situs Roboflow [12]. Situs Roboflow digunakan juga untuk proses *dataset augmentation* dan *export* dataset dengan format yang kompatibel pada masing-masing *framework* (YOLOv10 menggunakan *export format* YOLOv8 .txt dan EfficientDet menggunakan *export format* COCO .json).

Kondisi Normal | Mata Tertutup | Mulut Terbuka | Mata Tertutup dan Mulut Terbuka

Gambar 1. Pratinjau *dataset* yang akan digunakan

Proses pelabelan kondisi mata tertutup dan mulut terbuka dilakukan dengan menandai kedua mata yang tertutup sebagai satu buah label, serta pelabelan mulut yang terbuka dilakukan dengan menandai bagian sudut-sudut mulut pada objek pengemudi yang sedang menguap seperti yang ditunjukkan pada pratinjau sampel *dataset* di Gambar 1. Total *dataset* yang digunakan setelah melalui proses augmentasi adalah 14.568 buah *frame* dengan proporsi *Training:Validation:Test* sebesar 70%:20%: 10%. Di mana 70% proporsi pada data *training* dan 20% pada data *validation* merupakan implementasi *sample splitting* dengan cara memecah sebagian *dataset* yang digunakan untuk *training* sebagai data yang digunakan untuk dievaluasi setiap proses *epoch*-nya, di mana hal tersebut bisa membantu meminimalisir *total loss* pada fase *training* dan memperkuat kemampuan generalisasi model deteksi dengan pola yang bervariasi pada sebuah objek [13]. Proses *training* baik pada YOLOv10 maupun EfficientDet menggunakan konfigurasi yang sama, yakni dengan menggunakan varian *basic* dari kedua *framework* (YOLOv10n dan EfficientDet-D0) dengan jumlah *epochs* 25, *batch size* 16, *image size* 512px, *learning rate* 10^{-3}, dan jumlah *epochs* untuk *early stopping* sebanyak 3 buah *epochs* [5].



Gambar 2. Alur Evaluasi Performa Model Deteksi

Model deteksi yang telah selesai melakukan proses *training* akan dievaluasi dengan skema pengujian seperti yang divisualisasikan pada Gambar *2* terhadap 10% dari dataset yang digunakan, yakni 1447 buah *frame*. Baik YOLOv10 maupun EfficientDet melalui prosedur umum yang sama dalam evaluasinya, diawali dengan proses *setup* yang mengkonfigurasi *dependencies* dan mengunduh dataset dari situs Roboflow. Kemudian dilanjutkan dengan pemindahan *label ground truth* yang didapatkan dari hasil anotasi pada *dataset* ke sebuah variabel *list* yang beranggotakan *dictionary* yang merepresentasikan setiap data yang akan diolah. Data gambar akan dimuat melalui pengaksesan gambar pada *image path* setiap proses iterasi anggota dataset yang diolah. Setiap gambar yang diolah akan dilakukan proses perekaman hasil prediksi dan penyimpanan nilai *ground truth* di mana nilai tersebut dipetakan secara otomatis menggunakan *scikit learn*. Hasil dari *confusion matrix* yang dibuat oleh *scikit learn* kemudian diolah kembali menjadi matriks yang memetakan nilai *ground truth* dan *prediction* tanpa adanya nilai TN menggunakan *dataframe pandas* dan *library matplotlib* untuk melakukan visualisasi. Hasil akhir dari alur evaluasi performa model deteksi adalah gambar matriks yang menunjukkan perbandingan *instances ground truth* dan prediksi yang dilakukan, serta *prompt output* dari rerata kecepatan *inference* yang direkam. Nilai dari matriks yang dihasilkan akan digunakan untuk mengevaluasi keakuratan dengan membandingkan nilai *ground truth* dengan prediksi yang dilakukan oleh model setelah proses iterasi selesai.

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Hasil dari perbandingan *ground truth* dan prediksi model dipetakan dalam *confusion matrix* yang memberikan data nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) pada setiap kelasnya, nilai tersebut akan diolah menjadi data *Accuracy* dengan menggunakan Persamaan 1, *Precission* dengan menggunakan Persamaan 2 yang akan memberi gambaran akan

kemampuan model deteksi menangani deteksi *false positive*, dan *Recall* dengan menggunakan Persamaan 3 yang akan memberi gambaran akan kemampuan model deteksi menangani deteksi *false positive*.

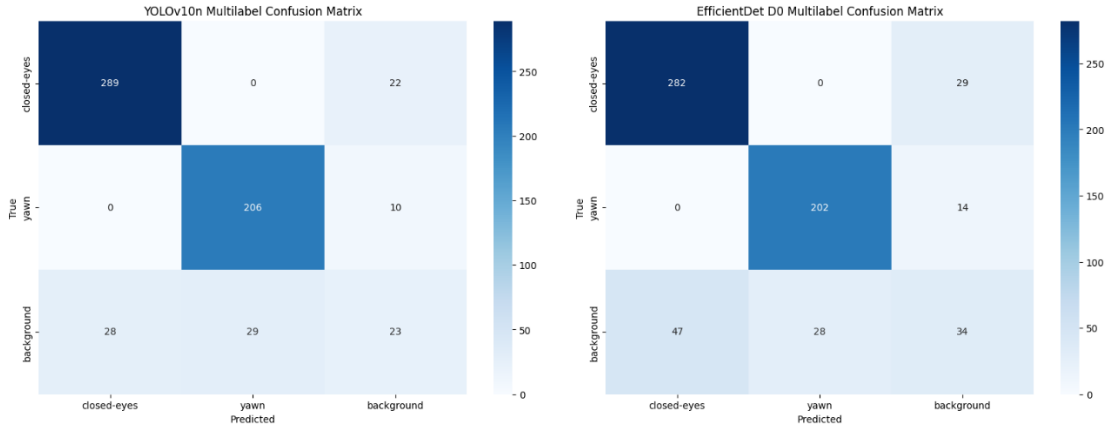
Evaluasi kecepatan deteksi pada setiap *framework* didapatkan dengan merata-ratakan hasil perekaman kecepatan deteksi pada setiap iterasi gambar yang diproses. Evaluasi kecepatan *inference* dilakukan pada perangkat yang sama dengan perangkat yang digunakan untuk proses *training*, yakni dengan menggunakan opsi perangkat GPU Nvidia T4 pada situs Google Colab.

3. Hasil dan Pembahasan

Proses *training* model deteksi dengan menggunakan proporsi sebanyak 10.218 data gambar untuk proporsi *training* dan 2.903 data gambar untuk *validation* berhasil dilakukan pada kedua *framework*. Pada YOLOv10, proses *training* berhenti pada *epoch* ke-25 dengan nilai total loss (akumulasi data *box loss* dan *regression loss*) akhir sebesar 1,168. Proses *training* model deteksi EfficientDet terhenti pada *epoch* ke-19 setelah *trigger early stopping* diaktifkan dengan nilai *total loss* terbaik pada 0,357.

Proses evaluasi keakuratan pada model deteksi dilakukan terhadap *test set* yang berjumlah 1447 *frames*, baik YOLOv10 maupun EfficientDet menggunakan *test set* yang sama untuk memastikan proses evaluasi model deteksi dapat dibandingkan karena menggunakan *test set* dengan jumlah dan gambar yang sama. Saat melakukan prediksi pada setiap gambar, kecepatan *inference* pada kedua *framework* direkam dengan merata-ratakan nilai selisih dari variabel waktu setelah dengan sebelum kode perintah "*prediction"* pada citra yang sedang diolah.

3.1. Evaluasi *Confusion Matrix*



Gambar 3 Hasil *confusion matrix* pada kedua *framework*

Data TP, TN, FP, dan FN pada masing-masing kelas di masing-masing *framework* yang telah didapatkan pada Gambar 3 kemudian diolah menjadi nilai *accuracy* untuk mendapatkan akurasi deteksi secara umum, nilai *precision* didapatkan untuk bisa memberi gambaran resiliensi model terhadap kesalahan deteksi *false positive*, dan *recall* yang nilainya didapatkan untuk memberi gambaran resiliensi model terhadap kesalahan *false negative* berdasarkan data uji.

Tabel 1. Hasil Evaluasi *Confusion Matrix Framework* YOLOv10 dan EfficientDet.

Framework	***Class***	***Accuracy* (%)**	***Precision* (%)**	***Recall* (%)**
YOLOv10n	*Closed Eyes*	91,76	91,17	92,93
	Yawn	93,57	87,66	95,37
EfficientDet-D0	*Closed Eyes*	88,05	85,71	90,68
	Yawn	93,40	87,83	93,52

Dari hasil perhitungan yang dipetakan pada Tabel 1, model deteksi, YOLOv10 lebih unggul pada seluruh aspek pengujian dibandingkan EfficientDet. Pada aspek akurasi dua buah kelas (dengan merata-ratakan metrik akurasi *closed eyes* dan *yawn*), YOLOv10 unggul 1,95% dengan rerata akurasi 92,67%. Pada aspek *precision*, YOLOv10 unggul sebesar 2,64% dengan rerata *precision* 89,41%. Pada aspek *recall*, YOLOv10 juga unggul sebesar 2,05% dengan rerata nilai *recall* sebesar 94,15%.

3.2. Evaluasi Kecepatan *Inference*

Perekaman kecepatan *inference* dihitung dengan merata-ratakan nilai selisih dari variabel waktu setelah dengan sebelum kode perintah "*prediction"* pada citra yang sedang diolah. Evaluasi kecepatan *inference* dilakukan oleh dua buah model deteksi terhadap 1447 gambar *test set* dan menghasilkan data *Avg. speed* yang dicantumkan pada Tabel 2.

Tabel 2. Perbandingan *Inference Speed Framework* YOLOv10 dan EfficientDet.

Framework	***Avg. speed* (ms)**
YOLOv10n	27,59
EfficientDet-D0	54,12

Didapatkan hasil rerata kecepatan *inference* pada YOLOv10 sebesar 27,59 ms dan 54,12 ms pada EfficientDet. Dari hasil evaluasi yang dilakukan, model deteksi dengan YOLOv10 secara signifikan mampu mendeteksi objek hingga dua kali lebih cepat dibandingkan dengan EfficientDet. Evaluasi ini memberikan gambaran awal mengenai bagaimana performa kecepatan deteksi pada *framework* yang digunakan saat diimplementasikan pada masa yang akan datang sebelum dipengaruhi komputasi lainnya seperti penyimpanan dan penanganan data, dan penambahan logika untuk memutuskan situasi pengemudi mengantuk yang akan mempengaruhi kecepatan deteksi secara umum.

3.3. Pembahasan

Berdasarkan pengembangan model dan proses evaluasi, baik YOLOv10 dan EfficientDet mampu melakukan deteksi dengan kinerja keakuratan lebih dari 85% dari seluruh metrik evaluasi yang diujikan, namun YOLOv10 sedikit lebih unggul pada seluruh aspek seperti yang dijelaskan pada subbab Evaluasi *Confusion Matrix*. Perbedaan signifikan terdapat pada hasil evaluasi kecepatan *inference* di mana YOLOv10 memiliki kecepatan *inference* hampir dua kali lebih cepat dibandingkan EfficientDet. Keunggulan pada kecepatan *inference* yang dimiliki YOLOv10 dapat memberikan keuntungan pada pengembangan selanjutnya, utamanya pada aspek yang sangat dipengaruhi oleh kecepatan *inference* seperti deteksi *slow blink* di mana sistem perlu mendeteksi situasi berkedip dengan durasi 400ms atau lebih untuk mendeteksi gejala mengantuk yang dialami pengemudi [14].

4. Kesimpulan

Mata terbuka dan mulut tertutup merupakan parameter utama yang dapat diidentifikasi dalam deteksi pengemudi mengantuk menggunakan citra digital sebagai data input. Penelitian ini mencakup pada perbandingan dua buah *framework single stage object detector* terkini saat diimplementasikan dalam pengembangan deteksi mata terbuka dan mulut tertutup. Perbandingan dilakukan dengan melakukan *training* menggunakan *dataset* yang sama dengan konfigurasi *hyperparameter framework* yang serupa pada masing-masing *framework*. Selanjutnya, dilakukan pengujian keakuratan dengan menghitung metrik *accuracy, precision,* dan *recall,* serta perekaman kecepatan *inference* masing-masing framework saat memproses 10% dari proporsi dataset yang digunakan. Perbedaan signifikan dari perbandingan yang dihasilkan pada penelitian ini adalah temuan pada kecepatan *inference* yang dihasilkan pada YOLOv10 hampir dua kali lebih cepat dibandingkan dengan EfficientDet, di mana YOLOv10 mampu melakukan pengolahan gambar dengan rata-rata kecepatan 27,59 ms, sementara EfficientDet memerlukan rata-rata waktu 54,12 ms untuk memproses *frame* yang diprediksi. Perbandingan kinerja keakuratan deteksi menghasilkan data evaluasi unggul pada *framework* YOLOv10 pada seluruh aspek, dengan selisih 1,95% lebih unggul pada aspek akurasi dengan nilai rerata 92,67%, 2,64% lebih unggul pada aspek *precision* dengan rerata *precision* 89,41%, serta selisih pada *recall* sebesar 2,05% dengan rerata nilai *recall* sebesar 94,15%. Keunggulan signifikan pada aspek kecepatan *inference* dapat memberi keuntungan pada pengembangan lebih lanjut yang membutuhkan kemampuan kecepatan deteksi dari model deteksi.

Pengembangan selanjutnya dapat dilakukan pada beberapa cabang yang cukup luas. Optimasi kecepatan *inference* pada arsitektur yang dimiliki EfficientDet dapat dilakukan untuk bisa meningkatkan kecepatan *inference*. Implementasi model deteksi YOLOv10 pada perangkat SBC terkini seperti seri perangkat Nvidia Jetson dapat dilakukan untuk bisa memperkecil *gap* pengembangan untuk diimplementasikan pada sistem ADAS. Serta pengamatan lebih mendalam pada *resource hardware* SBC yang digunakan saat proses *inference* juga dapat dilakukan untuk bisa mengamati bagaimana model deteksi yang telah dikembangkan menggunakan *resource* yang tersedia.

Daftar Pustaka

[1] "Jurnal Semester 1 Pusiknas Bareskrim Polri," Pusat Informasi Kriminal Nasional Bareskrim Polri, Jakarta, Feb. 2022.
[2] M. A. Rahman, S. Das, and X. Sun, "Understanding the drowsy driving crash patterns from correspondence regression analysis," *J. Safety Res.*, vol. 84, pp. 167–181, Apr. 2023.
[3] A. V Postolit, "Prospects for the Use of Artificial Intelligence and Computer Vision in Transport Systems and Connected Cars," *World Transp. Transp.*, vol. 19, no. 1, pp. 74–90, Jan. 2021.
[4] M. T. Soe, A. Zaw Min, H. T. Kyaw, M. Min Paing, S. M. Htet, and B. Aye, "Abnormal Behavior Detection in Real-time for Advanced Driver Assistance System (ADAS) using YOLO," in *2022 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 2022, pp. 1–6.
[5] R. Ayachi, M. Afif, Y. Said, and A. Ben Abdelali, "Drivers Fatigue Detection Using EfficientDet In Advanced Driver Assistance Systems," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2021, pp. 738–742.
[6] P. Inthanon and S. Mungsing, "Detection of Drowsiness from Facial Images in Real-Time Video Media using Nvidia Jetson Nano," in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020, pp. 246–249.
[7] J. D. Ortega *et al.*, "DMD: A Large-Scale Multi-modal Driver Monitoring Dataset for Attention and Alertness Analysis," *Springer Int. Publ.*, pp. 387–405, 2020.
[8] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: Yawning Detection Dataset." [object Object], 21-Apr-2020.
[9] H. Zhang and R. S. Cloutier, "Review on One-Stage Object Detection Based on Deep Learning," *EAI Endorsed Trans. e-Learning*, vol. 7, no. 23, p. 174181, 2022.
[10] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection." arXiv, 09-Jun-2024.
[11] M. Tan, R. Pang, and Q. V Le, "EfficientDet: Scalable and Efficient Object Detection." arXiv, 06-Jun-2020.
[12] D. B., N. J., and H. T., "Roboflow." 2024.
[13] Y. Bai *et al.*, "How Important is the Train-Validation Split in Meta-Learning?," *Proc. Mach. Learn. Res.*, vol. 139, pp. 543–553, 2021.
[14] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in *2010 International Conference on Machine and Web Intelligence (ICMWI)*, 2010, pp. 230–233.