

BAB V

SIMPULAN, IMPLIKASI, DAN REKOMENDASI

5.1. Simpulan

Berdasarkan hasil penelitian yang dibahas pada bagian bab temuan dan pembahasan didapatkan kesimpulan sebagai berikut.

- 1) Arsitektur YOLOv8n dapat digunakan sebagai algoritma *deep learning* pada model deteksi *speed bump* dengan mAP mencapai 92,5%;
- 2) Penggunaan arsitektur Mask RCNN dan YOLOv4 Tiny kurang tepat digunakan sebagai algoritma *deep learning* pada model deteksi *speed bump* karena model mengalami *overfitting* dan *underfitting*;
- 3) Model dapat melakukan pengukuran jarak *speed bump* pada jarak minimum 30 cm dan maksimum 20 meter;
- 4) Model deteksi dan estimasi jarak *speed bump* dapat diimplementasikan pada Jetson Nano dengan kecepatan deteksi sebesar 3-4 FPS jika menggunakan GPU;
- 5) Nilai *error* atau selisih antara jarak nyata dengan jarak estimasi yang diukur oleh model berada pada rentang 0,41-0,891 meter;
- 6) Proses deteksi dan pengukuran jarak *speed bump* dipengaruhi oleh intensitas cahaya. Semakin gelap pencahayaan di jalan maka model tidak dapat melakukan deteksi dan pengukuran jarak *speed bump*. Semakin terang pencahayaan di jalan maka model dapat melakukan deteksi dan pengukuran jarak *speed bump*. Semakin kurang pencahayaan maka semakin terbatas jarak *speed bump* yang dapat terdeteksi.

5.2. Implikasi

Berdasarkan kesimpulan di atas, arsitektur YOLOv8 memiliki performa yang lebih unggul dan memudahkan pengguna dalam melakukan *hyperparameter tuning*. Selain itu, YOLOv8 didukung oleh pihak ketiga dalam penyimpanan dan visualisasi *log* hasil pelatihan seperti salah satunya WandB sehingga dapat memudahkan pengguna dalam memantau performa model. Model deteksi dan estimasi jarak *speed bump* ini dapat diimplementasikan pada perangkat lain selain Jetson Nano. Model ini dapat memberikan kontribusi bagi perkembangan riset

deteksi objek pada kendaraan otonom. Selain itu, model ini dapat dikembangkan kembali, baik pada ranah pengembangan perangkat lunak, perangkat keras, maupun sistem tertanam. Konsep dari model deteksi dan estimasi jarak *speed bump* ini dapat digunakan pada objek lain selain *speed bump* dan pada lingkup lain selain lingkup kendaraan otonom.

5.3. Rekomendasi

Adapun rekomendasi mengenai perancangan model sistem deteksi dan estimasi jarak *speed bump* untuk kendaraan otonom menggunakan algoritma *deep learning* dengan YOLOv8 adalah sebagai berikut.

- 1) Pada saat pengambilan data, alangkah lebih baik jika langsung menggunakan kendaraan roda empat dibandingkan hanya dengan menggunakan *tripod*. Sehingga ketika malam hari bisa langsung menggunakan lampu kendaraan yang mana lampu kendaraan roda empat lebih terang dan lebih jauh jarak sorotnya dibandingkan dengan lampu kendaraan roda dua.
- 2) Pengambilan *dataset* dilakukan diberbagai lokasi yang berbeda sehingga *dataset* banyak dan beragam. Bisa ditambahkan kelas baru yaitu *speed bump* tidak bermarka.
- 3) Pengambilan data dan pengujian pada siang hari lebih baik dilakukan ketika cuaca tidak terlalu panas karena kamera Intel RealSense D455 memiliki sensor *infrared* sehingga sangat sensitif terhadap cuaca panas dan dapat mengakibatkan kamera *overheating*. Hal tersebut dapat mengakibatkan gambar atau video hasil perekaman berubah warna menjadi sedikit lebih gelap.
- 4) Perlu dilakukan *hyperparameter tuning* lebih lanjut ketika pelatihan model sehingga model dengan arsitektur YOLOv8n dapat mencapai mAP lebih dari 92,5%.
- 5) Pada proses estimasi jarak *speed bump* menggunakan fungsi pengukuran jarak *default* dari Intel Realsense SDK, rata-rata selisih antara jarak nyata dan jarak estimasi cukup besar, yaitu 0,41-0,891 meter. Oleh karena itu, untuk mengurangi selisih tersebut, dapat digunakan algoritma perhitungan jarak yang dirancang sesuai dengan kebutuhan atau melakukan kalibrasi ulang kamera Intel Realsense agar model dapat mengestimasi jarak dengan lebih presisi.

- 6) Pada tahapan implementasi, jika menggunakan Jetson Nano dengan dukungan GPU kecepatan deteksi model hanya mencapai 3-4 FPS. Oleh karena itu, untuk meningkatkan kecepatan deteksi model bisa menggunakan Jetson seri TX atau Jetson seri lainnya yang memiliki spesifikasi lebih tinggi dibanding dengan Jetson Nano.