**Pendeteksian Lubang Pada Jalan Menggunakan *Single Shot Multibox Detector* (SSD300-VGG16)**

**Abstrak -** Peningkatan jumlah kendaraan dapat menambah risiko terjadinya kemacetan lalu lintas dan kecelakaan di jalan. Kecelakaan tersebut umumnya dipicu oleh kesalahan pengemudi atau permasalahan terkait struktur jalan. Menurut data statistik, selama periode Januari-Mei 2021 terdapat 40 kasus kecelakaan lalu lintas akibat kondisi jalan salah satunya jalan berlubang. Lubang jalan adalah kondisi permukaan jalan di mana retak berkembang dan membentuk lubang yang memiliki diameter lebih dari 150 mm. Tahap krusial dalam perbaikan jalan adalah inspeksi. Di negara berkembang, seperti Indonesia, pengumpulan data dan identifikasi kerusakan jalan dilakukan secara manual dengan formulir survei atau inspeksi visual. Akan tetapi, metode ini memerlukan biaya tinggi, baik dari segi waktu maupun sumber daya manusia, dan cenderung tidak konsisten karena bergantung pada pengalaman individu surveyor. Untuk itu penelitian ini bertujuan untuk mengembangkan metode identifikasi jalan berlubang secara otomatis dengan pendekatan *machine learning*. Digunakan model SSD300-VGG16 karena kemampuannya dalam mendeteksi objek secara *real-time* dan objek dengan berbagai skala. Proses pelatihan melibatkan *transfer learning* pada dataset terbatas sebanyak 3770 gambar. Hasil penelitian menunjukkan bahwa model dengan *learning rate* 0.00001, ukuran *batch* 8, dan 40 *epoch* memberikan performa terbaik dengan mAP\_50 sebesar 68.3% dan waktu inferensi mencapai 30 ms per gambar ketika dijalankan menggunakan GPU.

Kata Kunci: SSD300-VGG16, *Machine Learning*, Deteksi Lubang Jalan

***Abstract*** *- The increasing number of vehicles can elevate the risk of traffic congestion and road accidents. These accidents are generally caused by driver errors or issues related to road structures. According to statistical data, during the period from January to May 2021, there were 40 cases of traffic accidents due to road conditions, including potholes. Potholes are conditions on the road surface where cracks develop, forming holes with a diameter of more than 150 mm. A crucial stage in road repair is inspection. In developing countries, such as Indonesia, data collection and road damage identification are currently done manually using survey forms or visual inspections. However, this method incurs high costs, both in terms of time and human resources, and tends to be inconsistent due to reliance on the individual experience of surveyors. Therefore, this research aims to develop an automated method for identifying potholes using a machine learning approach. The SSD300-VGG16 model is employed for its ability to detect objects in real-time and objects of various scales. The training process involves transfer learning on a limited dataset of 3770 images. The research results indicate that the model with a learning rate of 0.00001, batch size of 8, and 40 epochs performs the best with an mAP\_50 of 68.3%, and the inference time reaches 30 ms per image when run using GPU.*

*Keywords: SSD-VGG16, Machine Learning, Potholes Detection*

**PENDAHULUAN**

Bertambahnya jumlah kendaraan dapat meningkatkan risiko kemacetan lalu lintas dan kecelakaan di jalan. Kecelakaan umumnya disebabkan oleh kesalahan pengemudi atau masalah terkait struktur jalan seperti jalan rusak, jalan berlubang atau jalan licin. Berdasarkan data kecelakaan Dit Lantas Polda Metro Jaya periode Januari-Mei 2021 terdapat setidaknya 40 kasus kecelakaan lalu lintas akibat kondisi jalan, diantaranya 12 kasus akibat jalan rusak, 15 kasus akibat jalan berlubang, dan 13 kasus akibat faktor lainnya seperti tidak ada rambu dan tikungan tajam (Andika, 2021). Lubang jalan adalah area pada permukaan jalan di mana retak telah berkembang dan akhirnya membentuk lubang yang memiliki diameter lebih dari 150 mm (Ping et al., 2020). Jika tidak dilakukan perbaikan dan pengecekan secara berkala, maka lubang jalan yang ada akan semakin besar akibat cuaca buruk dan pergerakan kendaraan berat, yang tentunya akan sangat berbahaya bagi pengendara. Langkah yang sangat krusial dalam memperbaiki jalan adalah tahap inspeksi. Di negara berkembang termasuk Indonesia, pengumpulan data dan identifikasi kerusakan jalan dapat dilakukan secara manual dengan formulir survei yang diisi oleh surveyor atau dengan cara konvensional, di mana seseorang akan pergi ke lokasi, melakukan inspeksi visual, dan kemudian melakukan penilaian. Akan tetapi, kedua metode tersebut menimbulkan biaya yang tinggi, baik dari segi waktu maupun sumber daya manusia. Selain itu, proses penilaian menjadi tidak konsisten karena bergantung pada pengalaman individu dari surveyor atau operator (Ukhwah et al., 2019).

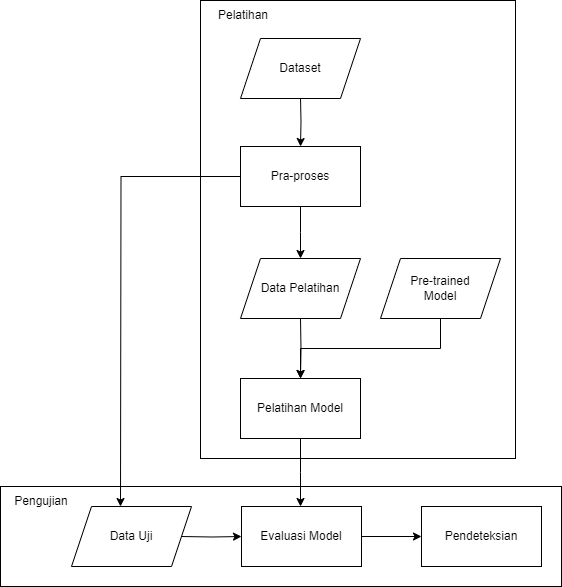
Dengan semakin berkembangnya teknologi, tugas pendeteksian lubang jalan dapat dilakukan secara otomatis. Salah satu pendekatannya adalah dengan menggunakan teknologi sensor untuk mendeteksi lubang jalan, seperti penelitian yang menggunakan jaringan sensor yang disebut BusNet, yang dipasang pada bus transportasi umum untuk memantau kondisi permukaan jalan (De Zoysa et al., 2007). Lalu penelitian lainnya yang memanfaatkan *accelerometer smartphone* untuk memonitor permukaan jalan dengan menganalisis sinyal getaran vertikal. Meskipun dapat mendeteksi kejadian abnormal seperti lubang jalan pada berbagai kecepatan kendaraan, pendekatan ini tidak dapat digunakan untuk memprediksi lubang jalan secara *real-time*, karena *accelerometer* perlu merekam getaran saat kendaraan sudah melewati lubang (Harikrishnan & Gopi, 2017). Pendekatan lainnya adalah dengan menggunakan teknik *machine learning* dan pemrosesan citra digital seperti pada penelitian dalam mendeteksi jalan berlubang menggunakan SSD-Mobilenet pada tiga macam variasi dataset yaitu dataset normal, *dashboard* dan closeup. Hasil penelitian menunjukkan nilai mAP terbaik sebesar 0.69 pada IoU 0.5 (Pakpahan & Dewi, 2021). Akan tetapi, model SSD-Mobilenet ini menggunakan arsitektur yang tidak begitu kompleks karena dirancang untuk ukuran yang lebih kecil dan hemat sumber daya, ideal untuk perangkat dengan keterbatasan sumber daya komputasi, seperti perangkat bergerak, sehingga akurasi bukanlah yang terbaik (Kang, 2020) .

Dengan semakin berkembangnya teknologi, masalah sumber daya komputasi sudah semakin dapat diatasi. Sehingga dalam penelitian ini, penulis mengusulkan solusi pendeteksian jalan berlubang secara otomatis dengan menggunakan algoritma *machine learning* SSD (Single-shot Multibox Detector) dengan arsitektur yang lebih kompleks. Dilakukan metode *transfer learning* pada penelitian ini, karena jumlah dataset yang digunakan relatif sedikit yaitu hanya sebanyak 3770 gambar. Sehingga model yang dipilih adalah SSD300-VGG16 yang telah dilakukan *pretrain* yang tersedia pada PyTorch. Model ini menggunakan arsitektur VGG16 sebagai bagian dari *backbone* yang nantinya bertugas untuk mengekstraksi fitur dari gambar inputan (Liu et al., 2016). Model ini dipilih karena dapat mendeteksi objek secara *real-time*. Sebagai contoh, model SSD512 dapat menghasilkan akurasi sebesar 0,63 mAP dengan kecepatan deteksi 28 fps, sementara model SSD300 menghasilkan akurasi 0,57 mAP dengan kecepatan deteksi 51 fps (Ramadhan, 2021). Selain itu model SSD juga baik dalam mendeteksi objek dalam berbagai ukuran karena SSD menggunakan berbagai lapisan peta fitur untuk deteksi objek, yang memungkinkan deteksi objek pada berbagai skala (Chen et al., 2019). Sehingga dengan kemampuan ini, diharapkan model nantinya dapat diintegrasikan dengan teknologi pada *dashcam* mobil petugas sebagai langkah alternatif untuk inspeksi jalan berlubang. Selain itu, perusahaan dalam industri transportasi juga dapat mengimplementasikan sistem ini pada kendaraan mereka, yang akan membantu mengurangi jumlah kecelakaan dan kerusakan mobil yang disebabkan oleh lubang jalan.

Selanjutnya jurnal ini disusun sebagai berikut: Bagian II menyajikan studi literatur yang relevan terkait dengan deteksi lubang jalan. Bagian III menjelaskan solusi yang diusulkan beserta rancangannya. Pada Bagian IV, hasil simulasi dan analisis kinerja disajikan. Terakhir, kesimpulan diberikan pada Bagian V.

**METODE PENELITIAN**

Langkah-langkah yang dijalankan dalam penelitian ini dimulai dengan menyiapkan dataset yang terdiri dari gambar dan anotasi sebagai labelnya. Setelah itu, data yang telah disiapkan dibagi menjadi tiga bagian, yaitu data latih, data validasi, dan data uji. Data latih yang berupa gambar dan target atau *bounding box* digunakan untuk melakukan *fine-tuning* terhadap model *pre-train* SSD300-VGG16. Hasil dari tahap pemodelan ini berupa model terlatih. Evaluasi kinerja model dilakukan dengan melakukan deteksi menggunakan data uji. Model dengan performa terbaik akan disimpan bobotnya dan digunakan untuk melakukan inferensi. Rancangan metode ini dapat dilihat pada Gambar x.x.



Gambar x.x Diagram rancangan metode

*Dataset*

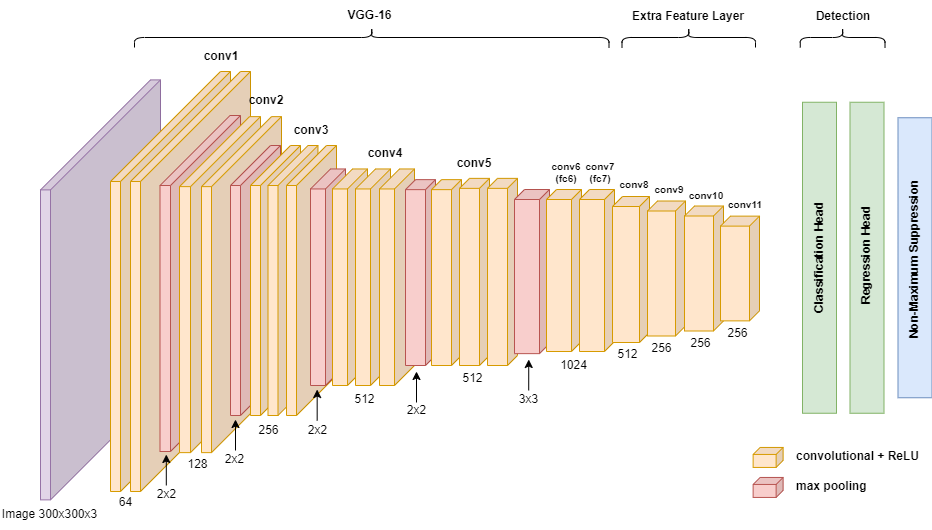
Dalam penelitian ini, citra jalan didapat melalui sumber sekunder yaitu Roboflow yang berisi total 3770 data. Data terdiri dari data gambar jalan berlubang dengan berbagai ukuran dan bentuk berdimensi 640 x 640 piksel dengan ekstensi jpg dan data anotasi berekstensi xml dengan format Pascal VOC yang berisi informasi diantaranya mengenai dimensi dari gambar dan koordinat *bounding box* terhadap gambar dengan format x\_min, y\_min, x\_max dan y\_max. Setelah dilakukan praproses, data yang digunakan hanya sebanyak 3439 gambar. Dataset ini dibagi menjadi data latih (70%) dengan 2407 gambar, data validasi (20%) dengan 688 gambar, dan data uji (10%) dengan 344 gambar.

*Pra-proses*

Pelatihan model SSD300 memerlukan input berupa tensor dan target (berupa *list dictionary*) yang menyimpan informasi ground-truth (Liu et al., 2016). Dibutuhkan pra-pemrosesan untuk mendapatkan *ground-truth* untuk setiap gambar. Setiap data anotasi akan diuraikan (*parsed*) untuk mendapatkan informasi koordinat dari *bounding-box*. Nilai piksel pada setiap tensor gambar juga harus berada dalam rentang 0 hingga 1. Oleh karena itu, diperlukan normalisasi agar nilai pikselnya berada dalam rentang yang diinginkan. Selain itu, perlu dilakukan pengecekan untuk koordinat *bounding-box*nya. Jika koordinat melebihi lebar atau tinggi dari gambar, maka data tidak akan digunakan karena akan menyebabkan kesalahan saat melakukan pelatihan.

Pada penelitian ini, dilakukan eksperimen untuk mengevaluasi pengaruh augmentasi data terhadap performa model. Oleh karena itu, dilakukan pelatihan pada dua set data, yaitu satu set data tanpa menggunakan augmentasi dan satu set data dengan penerapan augmentasi. Augmentasi data adalah teknik yang dimanfaatkan untuk meningkatkan jumlah data dengan memodifikasi data yang sudah ada berupa transformasi geometri, penyesuaian kontras, dan lainnya. Salah satu tujuan dari augmentasi data adalah mencegah terjadinya overfitting dengan membuat model lebih adaptif terhadap variasi kecil (Setiawan, 2021). Pada penelitian ini augmentasi yang dipilih didasarkan pada hasil penelitian sebelumnya dengan hasil model dengan performa terbaik dilatih pada data yang melakukan augmentasi berupa variasi transformasi, variasi warna, dan variasi ukuran (Liu et al., 2016). Sehingga dilakukan augmentasi data, termasuk horizontal flip, *ShiftScaleRotate* untuk melakukan pergeseran (*shift*), perubahan skala (*scale*), dan rotasi, peningkatan kecerahan dan kontras, penambahan *Gaussian Noise* untuk mensimulasikan kondisi lingkungan tertentu, dan *RandomResizedCrop* dengan ukuran 480 x 480 sesuai dengan ukuran input model selama pelatihan.

*Model*



Gambar x.x Arsitektur SSD300-VGG16 (Sumber: [[2]](https://arxiv.org/pdf/1512.02325.pdf))

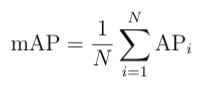
Pada penelitian ini digunakan model arsitektur SSD300-VGG16, di mana model ini menggabungkan dua komponen utama yaitu VGG16 sebagai *backbone* yang berfungsi untuk melakukan ekstraksi fitur dan jaringan SSD sebagai *head* yang berfungsi sebagai *object detector*. VGG16 adalah salah satu arsitektur model jaringan saraf konvolusional (Convolutional Neural Network atau CNN). VGG16 memiliki arsitektur yang sangat dalam dengan 16 lapisan konvolusi yang menggunakan filter 3 x 3 dan lapisan-lapisan lainnya seperti pooling dan fully connected. Aktivasi ReLU (*Rectified Linear Unit*) digunakan di seluruh jaringan, kecuali pada lapisan output yang menggunakan aktivasi softmax (Simonyan & Zisserman, 2015). SSD (Single Shot Multibox Detector) menggunakan lapisan konvolusi VGG16 untuk menghasilkan *multiple feature maps* dengan resolusi berbeda. Setiap feature map merepresentasikan gambar pada skala tertentu. Pada setiap feature map, SSD menempatkan kotak-kotak default dengan skala dan rasio aspek tertentu, berfungsi sebagai kandidat deteksi objek. Setiap kotak default digunakan untuk prediksi lokalisasi bounding box (koordinat x, y, lebar, tinggi) dan *confidence* dari objek. Dengan menggabungkan langkah-langkah tradisional deteksi objek, SSD memungkinkan deteksi dalam satu tahap (single shot), menjadikannya efisien (Liu et al., 2016).

*Pelatihan Model*

Pada penelitian ini digunakan pendekatan transfer learning di mana model yang telah dilatih pada tugas sumber digunakan sebagai titik awal untuk melatih model pada tugas target atau disebut juga model pre-train (Ruder, 2017) . Lalu digunakan teknik *fine-tuning* dimana model *pre-train* ini akan dilatih menggunakan dataset pada penelitian untuk menyesuaikan dengan tugas pendeteksian lubang jalan. Sebelum memulai pelatihan, perlu dilakukan konfigurasi agar sesuai dengan dataset. Misalnya, mengganti jumlah channel input untuk lapisan kepala (*head)* model dengan dimensi input (480, 480), sesuai dengan dimensi gambar inputan yang berukuran 480 x 480. Selain itu, perlu mengganti jumlah kelas agar sesuai dengan dataset, yakni sebanyak 2 kelas termasuk kelas *background*. Pelatihan dilakukan pada berbagai macam variasi hyperparameter untuk mencari performa model terbaik. Pemilihan hyperparameter dilakukan secara manual. Hyperparameter yang dikonfigurasi diantaranya *learning rate*, ukuran *batch*, dan banyaknya *epoch*. Digunakan Adam optimizer sebagai algoritma optimasi selama pelatihan. Lalu selama pelatihan digunakan juga *learning rate scheduler* untuk memantau perubahan dari *learning rate* hal ini dilakukan untuk menghindari risiko bahwa perubahan learning rate dapat menyebabkan model menuju ke arah yang tidak menguntungkan dengan cara memuat bobot terbaik yang sebelumnya telah disimpan.

*Evaluasi Model*

Pengujian model dilakukan dengan tujuan untuk mengevaluasi kinerja model dalam mengidentifikasi keberhasilan mendeteksi lubang. Nilai evaluasi didapat dari hasil model melakukan prediksi terhadap data uji. Hasil dari prediksi akan dilakukan evaluasi dengan menggunakan metrik pengukuran *mean average precision* (mAP). Metrik evaluasi ini dipilih karena merupakan standar dalam menganalisis akurasi dari sebuah model deteksi objek. *Mean average precision* (mAP) menggabungkan beberapa konsep metrik evaluasi seperti *confusion matrix*, *precision*, *recall*, serta IoU (*Intersection over Union*) untuk mengukur sejauh mana *bounding box* hasil prediksi model tumpang tindih dengan *ground truth* (Shah, 2022). Pada deteksi objek, nilai *precision* didapat dengan melihat hasil IoU terhadap *threshold*, jika IoU berada di bawah *threshold* maka diklasifikasikan sebagai *false positive* dan bila sebaliknya diklasifikasikan sebagai *true positive*. Nilai mAP didapat dengan menghitung rata-rata dari nilai AP pada setiap kelas (Gad, 2022). Pada kasus pendeteksian yang hanya memiliki satu kelas seperti pada penelitian ini, nilai mAP sama dengan nilai AP.



Gambar x.x Rumus mAP (sumber : [[2]](https://www.v7labs.com/blog/mean-average-precision))

Pada penelitian ini dipilih mAP\_50 dan mAP\_50\_95 yaitu untuk mengukur kinerja model deteksi objek pada ambang batas IoU sebesar 0.5 dan IoU antara 0.5 hingga 0.95. *Train loss* dan *validation loss* juga digunakan sebagai evaluasi untuk mendeteksi overfitting dan konvergensi dari model.

**HASIL DAN PEMBAHASAN**

Dalam penelitian ini, pengujian dilakukan di lingkungan Google Colab dengan memanfaatkan GPU Tesla T4 dengan kapasitas 15 GB dan CUDA 12.2 untuk mempercepat proses komputasi. Pengembangan model dilakukan menggunakan bahasa pemrograman Python dan framework PyTorch.

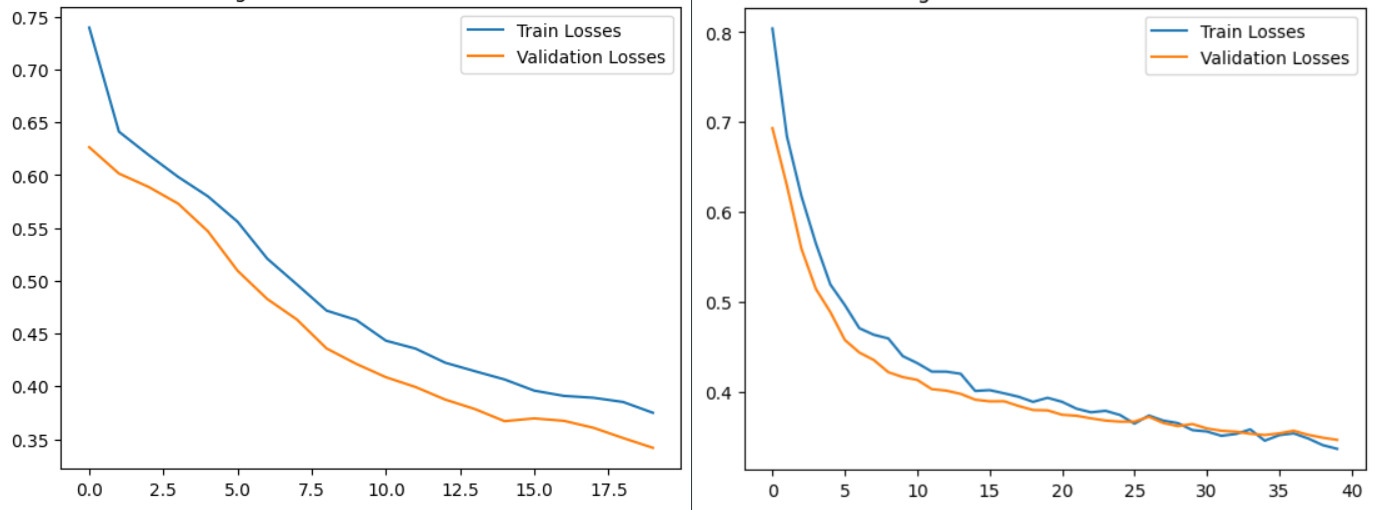
1. Evaluasi mAP dan *loss*

Sebelum melakukan evaluasi, perlu dilakukan optimalisasi untuk hasil prediksinya misalnya dengan mempertahankan hasil prediksi yang hanya melebihi nilai *threshold* yang ditentukan yaitu sebesar 0.25. Selain itu, dilakukan juga teknik *Non-Maximum Suppression* (NMS) untuk mengeliminasi *bounding box* yang tumpang tindih (*overlap*) pada objek yang sama dengan pengaturan nilai *threshold* sebesar 0.75. Berikut merupakan hasil evaluasi berupa nilai mAP untuk tiap variasi *hyperparameter* yang digunakan.

| **Model** | **Learning Rate** | **Batch** | **Epoch** | **mAP\_50** | **mAP\_50\_95** | **Training Loss** | **Validation Loss** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.001 | 4 | 20 | 0.01 | 0.002 | 1.12 | 1.171 |
| 2 | 0.0001 | 4 | 20 | 0.31 | 0.176 | 0.632 | 0.652 |
| 3 | 0.00001 | 4 | 40 | 0.302 | 0.195 | 0.495 | 0.556 |
| 4 | 0.00001 | 4 | 20 | 0.365 | 0.201 | 0.591 | 0.576 |
| 5 | 0.00001 | 8 | 20 | 0.633 | 0.364 | 0.374 | 0.341 |
| 6 | 0.00001 | 8 | 40 | 0.683 | 0.424 | 0.336 | 0.346 |
| 7 | 0.00001 | 8 | 50 | 0.646 | 0.404 | 0.324 | 0.325 |
| 8 | 0.00001 | 10 | 40 | 0.303 | 0.178 | 0.282 | 0.278 |
| 9 | 0.000001 | 4 | 20 | 0.102 | 0.028 | 1.23 | 1.184 |

Tabel x.x Hasil pengujian variasi *hyperparameter*

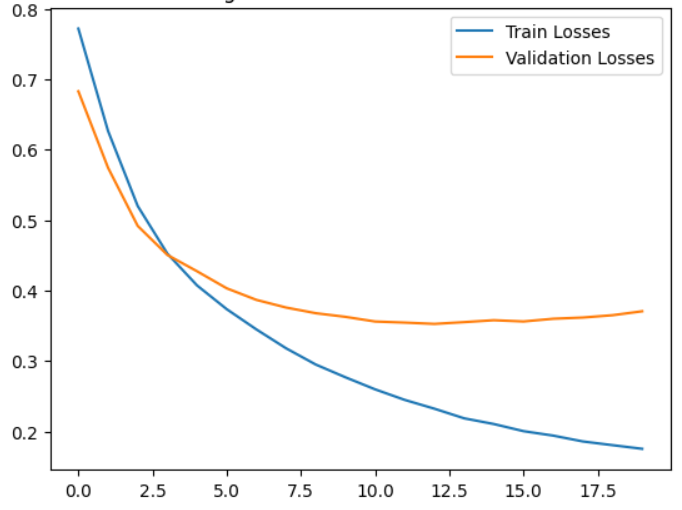
Dapat dilihat dari hasil di atas, model 6 dengan *learning rate* 0.00001, ukuran *batch* 8 dan jumlah *epoch* sebanyak 40 adalah model dengan performa terbaik yaitu mendapatkan skor mAP\_50 sebesar 0.683 atau 68.3%. Akan tetapi, didapatkan bahwa nilai *loss* tidak selalu menggambarkan bahwa model tersebut baik, seperti yang terlihat pada model 8 memiliki nilai *loss* yang lebih baik daripada model 6 tapi berbanding terbalik dengan nilai mAPnya. Hal ini mungkin disebabkan oleh peningkatan ukuran *batch*. Dengan ukuran batch yang lebih besar, model dapat memiliki kecenderungan untuk lebih cepat "menghafal" data pelatihan, yang dapat mengarah pada *overfitting*. Ini dapat menyebabkan peningkatan dalam loss pelatihan tetapi mungkin tidak mencerminkan peningkatan dalam kemampuan model untuk melakukan generalisasi pada data uji. Penambahan jumlah epoch terbukti dapat meningkatkan skor mAP ketika model belum mencapai konvergensi seperti yang terlihat pada gambar x.x.



Gambar x.x Grafik *loss* model 5 dan model 6

Dapat dilihat bahwa saat epoch sebanyak 20 tidak terjadi *overfitting* dan grafik masih cenderung menurun menandakan belum terjadi konvergensi. Lalu dilakukan penambahan epoch menjadi sebanyak 40 dan grafik mulai menunjukkan konvergensi. Sekilas terlihat dari nilai *loss* tidak terdapat perubahan tetapi saat dilakukan evaluasi menggunakan mAP terdapat perubahan yang cukup signifikan.

2. Augmentasi Data



Gambar x.x Grafik *loss* dataset tanpa augmentsi

Gambar x.x menunjukkan bahwa penggunaan augmentasi terhadap dataset dapat mencegah *overfitting*. Ketika model dilatih dengan konfigurasi *learning rate* sebesar 0.00001, *batch size* 8, dan *epoch* sebanyak 20 menggunakan dataset tanpa augmentasi, terlihat adanya *overfitting* pada *epoch* ke-5. *Overfitting* ditandai dengan penurunan *train loss* yang terus-menerus, sementara *validation loss* cenderung datar atau bahkan mulai meningkat. Nilai mAP yang didapat juga kurang baik yaitu sebesar 0.302 untuk IoU 0.5 dan 0.180 untuk IoU 0.5 hingga 0.95.

3. Inferensi

Selanjutnya merupakan hasil prediksi model terhadap data uji.

| **Ground Truth** | **Hasil Prediksi** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Tabel x.x Hasil prediksi data uji

Dapat dilihat model mampu memprediksi dengan baik setiap *ground truth*, hanya terdapat perbedaan pada ukuran dari *bounding box*. Selanjutnya dilakukan prediksi pada data baru yang tidak digunakan selama pelatihan atau yang disebut dengan proses inferensi.

Gambar x.x Hasil inferensi gambar

Model mampu mendeteksi objek tetapi untuk objek yang berada pada posisi yang jauh atau berukuran kecil model belum dapat mendeteksinya dengan baik. Hal ini disebabkan karena kurangnya representasi gambar dengan objek kecil atau objek yang berjarak jauh, sehingga model mungkin tidak belajar dengan baik untuk mengenali situasi tersebut. Waktu inferensi yang didapat untuk satu gambar adalah sekitar 30 ms ketika dijalankan menggunakan GPU dan 1500 ms ketika dijalankan tanpa menggunakan GPU. Ketika melakukan inferensi terhadap video mendapatkan kecepatan inferensi sebesar 60 fps.

**KESIMPULAN**

Secara keseluruhan, penelitian ini mencoba mengatasi masalah inspeksi jalan berlubang yang masih dilakukan secara manual. Solusi yang diusulkan, yaitu pendeteksian otomatis lubang jalan menggunakan algoritma *machine learning* SSD300-VGG16. Model yang dikembangkan dengan metode *transfer learning* pada dataset terbatas mampu memberikan performa terbaik dengan mAP\_50 mencapai 68.3%. Penggunaan dataset dari sumber sekunder dengan augmentasi data juga terbukti efektif dalam mencegah *overfitting*. Meskipun model mampu melakukan inferensi pada gambar dan video, beberapa keterbatasan seperti deteksi objek pada jarak jauh atau berukuran kecil masih perlu diperbaiki. Waktu inferensi yang relatif cepat, mencapai 30 ms per gambar, menunjukkan potensi implementasi solusi ini pada teknologi *dashcam* mobil. Untuk penelitian selanjutnya, dapat dieksplorasi variasi *hyperparameter* lainnya agar nilai mAP dapat lebih baik lagi. Selain itu, dapat digunakan pendekatan *thermal imaging* agar model dapat mendeteksi objek pada tingkat pencahayaan yang minim seperti pada kondisi malam hari.

**REFERENSI**

Andika, M. L. (2021). *Duh! Jalan Rusak Jadi Penyebab Banyak Kecelakan Lalu Lintas di Jakarta*.DetikOto.Com.https://oto.detik.com/berita/d-5624401/duh-jalan-rusak-jadi-penyebab-banyak-kecelakan-lalu-lintas-di-jakarta

Chen, Z., Wu, K., Li, Y., Wang, M., & Li, W. (2019). SSD-MSN: An Improved Multi-Scale Object Detection Network Based on SSD. *IEEE Access*, *7*, 80622–80632. https://doi.org/10.1109/ACCESS.2019.2923016

De Zoysa, K., Keppitiyagama, C., Seneviratne, G. P., & Shihan, W. W. A. T. (2007). A public transport system based sensor network for road surface condition monitoring. *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions, NSDR’07*. https://doi.org/10.1145/1326571.1326585

Gad, A. F. (2022). *Evaluating Object Detection Models Using Mean Average Precision (mAP)*. https://blog.paperspace.com/mean-average-precision/amp/

Harikrishnan, P. M., & Gopi, V. P. (2017). Vehicle Vibration Signal Processing for Road Surface Monitoring. *IEEE Sensors Journal*, *17*(16), 5192–5197. https://doi.org/10.1109/JSEN.2017.2719865

Kang, Y. (2020). Research on SSD base network. *IOP Conference Series: Materials Science and Engineering*, *768*(7). https://doi.org/10.1088/1757-899X/768/7/072031

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9905 LNCS*, 21–37. https://doi.org/10.1007/978-3-319-46448-0\_2

Pakpahan, I. B., & Dewi, I. C. (2021). Pendeteksian Lubang Pada Jalanan Menggunakan Metode SSD-MobileNet. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, *11*(2), 213. https://doi.org/10.22146/ijeis.60157

Ping, P., Yang, X., & Gao, Z. (2020). A Deep Learning Approach for Street Pothole Detection. *Proceedings - 2020 IEEE 6th International Conference on Big Data Computing Service and Applications, BigDataService 2020*, 198–204. https://doi.org/10.1109/BigDataService49289.2020.00039

Ramadhan, M. H. A. (2021). *Pendeteksian Jenis Sampah Dengan Single Shot Multibox Detector Secara Real Time*. https://repository.unsri.ac.id/61053/

Ruder, S. (2017). Transfer Learning - Machine Learning’s Next Frontier. *PPT*, 1–31. https://www.ruder.io/transfer-learning/

Setiawan, A. (2021). *Augmentasi Data Pada Computer Vision*. Medium.Com. https://medium.com/data-folks-indonesia/augmentasi-data-pada-computer-vision-45c5ebe10e8f

Shah, D. (2022). *Mean Average Precision (mAP) Explained: Everything You Need to Know*. V7labs.Com. https://www.v7labs.com/blog/mean-average-precision

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Ukhwah, E. N., Yuniarno, E. M., & Suprapto, Y. K. (2019). Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network. *Proceedings - 2019 International Seminar on Intelligent Technology and Its Application, ISITIA 2019*, 35–40. https://doi.org/10.1109/ISITIA.2019.8937176