

INTISARI

OPTIMASI YOLOV8 UNTUK SEGMENTASI JENIS PERKERASAN JALAN MELALUI PENGURANGAN PARAMETER MENGGUNAKAN GHOSTCONV DAN C3GHOST

Oleh

Edy Pratamajaya

23/525999/PPA/06629

Model *YOLOv8* dikenal memiliki performa unggul dalam tugas segmentasi instance, khususnya pada aspek akurasi. Namun, penggunaan lapisan konvolusional standar menyebabkan tingginya kompleksitas komputasi, sehingga kurang ideal untuk perangkat dengan keterbatasan sumber daya. Untuk mengatasi hal tersebut, dilakukan modifikasi arsitektur dengan mengganti sebagian lapisan konvolusional menggunakan *GhostConv* dan *C3Ghost*, yang menghasilkan model baru bernama *YOLO-GhostConv+C3Ghost*. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa *YOLOv8* dan *YOLO-GhostConv+C3Ghost* dari segi akurasi segmentasi, efisiensi komputasi, serta kecepatan inferensi.

Berdasarkan hasil pengujian, konfigurasi terbaik *YOLOv8* (menggunakan AdamW, *learning rate* 0.001) mencatat *precision* hingga 0,976 dan mAP-50 sebesar 0,957 untuk prediksi box, serta *precision* 0,976 dan mAP-50 0,954 untuk prediksi mask. Sementara itu, *YOLO-GhostConv+C3Ghost* pada konfigurasi terbaiknya (Adam, *learning rate* 0.005) mencatat *precision* box tertinggi 0,988 dan mAP-50 0,958, serta *precision* mask 0,988 dan mAP-50 0,959. Namun, *YOLO-GhostConv+C3Ghost* memiliki keunggulan lebih menonjol dalam efisiensi, dengan waktu inferensi rata-rata sebesar 3,4–4,2 ms dan waktu pascapemrosesan 5,1–7,6 ms, dibandingkan *YOLOv8* yang memiliki waktu inferensi 3,5–4,2 ms dan pascapemrosesan 4,4–6,0 ms. Model *YOLO-GhostConv+C3Ghost* juga menunjukkan waktu pra-pemrosesan lebih singkat pada beberapa konfigurasi, yakni hingga 0,3–0,8 ms.

Eksperimen juga menunjukkan bahwa penggunaan optimizer AdamW memberikan stabilitas pelatihan yang lebih baik dibandingkan Adam, meskipun Adam dapat meningkatkan *precision* dengan potensi penurunan recall. Penggunaan *learning rate* sebesar 0,005 menghasilkan kinerja optimal pada konfigurasi tertentu, namun cenderung memperlambat waktu inferensi pada beberapa kasus.

Dengan demikian, integrasi *GhostConv* dan *C3Ghost* pada *YOLO-Ghost* terbukti dapat meningkatkan efisiensi komputasi tanpa mengorbankan akurasi

secara signifikan. *YOLO-Ghost* direkomendasikan untuk implementasi pada perangkat dengan keterbatasan daya komputasi, sementara *YOLOv8* tetap menjadi pilihan utama pada aplikasi yang membutuhkan akurasi dan presisi tinggi secara konsisten.

Kata Kunci: *YOLOv8, YOLO-Ghost, segmentasi instance, efisiensi komputasi, kecepatan inferensi, GhostConv, C3Ghost, mAP, FPS, FLOPs.*

ABSTRACT

OPTIMIZATION OF YOLOV8 FOR PAVEMENT TYPE SEGMENTATION THROUGH PARAMETER REDUCTION USING GHOSTCONV AND C3GHOST

Oleh

Edy Pratamajaya
23/525999/PPA/06629

The YOLOv8 model has demonstrated strong performance in instance segmentation tasks, particularly in terms of detection accuracy. However, its reliance on standard convolutional layers leads to high computational complexity, making it less suitable for deployment on resource-constrained devices. To address this limitation, architectural modifications were introduced by replacing certain convolutional layers with GhostConv and C3Ghost modules, resulting in a new variant called YOLO-Ghost. This study aims to analyze and compare the performance of YOLOv8 and YOLO-Ghost in terms of segmentation accuracy, computational efficiency, and inference speed.

Experimental results indicate that the best-performing YOLOv8 configuration (AdamW optimizer, learning rate 0.001) achieved a box prediction precision of 0.976 and mAP-50 of 0.957, along with mask prediction precision of 0.976 and mAP-50 of 0.954. In contrast, the optimal YOLO-Ghost configuration (Adam optimizer, learning rate 0.005) attained a box precision of 0.988 and mAP-50 of 0.958, with mask precision of 0.988 and mAP-50 of 0.959. Furthermore, YOLO-Ghost demonstrated superior computational efficiency, with average inference times ranging from 3.4 to 4.2 ms and post-processing times between 5.1 and 7.6 ms, compared to YOLOv8's 3.5 to 4.2 ms inference and 4.4 to 6.0 ms post-processing times. YOLO-Ghost also exhibited faster pre-processing in several configurations, with durations as low as 0.3–0.8 ms.

In addition, optimization experiments showed that AdamW provided more stable training performance than Adam, although the latter improved precision at the cost of reduced recall. A learning rate of 0.005 yielded optimal performance in certain settings but occasionally led to slower inference times.

In conclusion, the integration of GhostConv and C3Ghost into the YOLO-Ghost architecture significantly enhances computational efficiency without substantial degradation in accuracy. YOLO-Ghost is therefore more suitable for deployment on low-resource systems, while YOLOv8 remains preferable for applications requiring consistently high segmentation precision.

Keywords: *YOLOv8, YOLO-Ghost, instance segmentation, computational efficiency, inference speed, GhostConv, C3Ghost, mean Average Precision (mAP), frame per second (FPS), FLOPs.*