

Tugas 2: Observasi Berbagai Arsitektur Terkini Pada Sistem Deteksi Objek

Link source code : [Google Drive](#)

Dataset yang digunakan dalam tugas ini adalah *Hand Gesture Dataset*, yang terdapat 15 kelas gestur tangan. Sebelas kelas tersebut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, addition, division, equal, minus, dan multiplication . Dataset dibagi ke dalam 3 bagian: train, validation, dan test.

A. Arsitektur yang Digunakan

1. YOLOv5n

YOLOv5n adalah varian ringan (*nano*) dari arsitektur YOLOv5 yang didesain untuk deteksi objek secara cepat dan akurat. YOLO menggunakan satu jaringan konvolusi untuk mendeteksi objek di seluruh gambar dalam satu proses (*single pass*). Arsitektur ini dibangun dengan *convolution layers* yang lebih dalam dan dioptimalkan untuk kecepatan inferensi yang lebih tinggi.

Cara kerja: Dalam dataset ini, YOLOv5n membagi gambar menjadi grid dan memprediksi objek serta bounding box dalam satu langkah. Deteksi dilakukan dengan mengenali ciri khas objek seperti bentuk atau warna pada setiap grid, misalnya bentuk atau tekstur pada jari.

2. SSD MobilenetV3

SSD adalah arsitektur deteksi objek yang bekerja dengan cara memprediksi *bounding box* dan kategori objek secara bersamaan dalam satu langkah. MobileNetV3 adalah backbone untuk SSD yang memberikan keseimbangan antara akurasi dan kecepatan dengan ukuran model yang kecil. SSD menggunakan *convolution layers* untuk menghasilkan prediksi objek pada berbagai skala.

Cara kerja: SSD MobileNetV3 mendeteksi objek dengan memprediksi *bounding box* dan kategori objek pada berbagai skala dalam gambar. Menggunakan konvolusi untuk mengenali objek, seperti pola atau bentuk jari.

3. RT-DETR

RT-DETR adalah varian dari DETR (*Detection Transformer*), yang mengintegrasikan kemampuan transformer untuk menangani tugas deteksi objek. RT-DETR mengoptimalkan arsitektur DETR untuk meningkatkan kecepatan inferensi dan akurasi dengan menggunakan teknik pemrosesan yang lebih efisien. Arsitektur ini memanfaatkan *attention mechanism* untuk memahami hubungan antara bagian-bagian gambar dan mengidentifikasi objek dengan cara yang lebih kontekstual.

Cara kerja: RT-DETR menggunakan transformer untuk mendeteksi objek dengan memperhatikan hubungan antar bagian gambar. Mengidentifikasi objek dengan menganalisis konteks keseluruhan gambar, seperti susunan dan kelompok objek.

B. Perbandingan Performa

Model	mAP@0.5	mAP@0.75	mAP@0.5 0.95	mAP overall
YOLOv5n	0.9474	0.8678	0.7546	Tidak tersedia
SSDMobileNetV3	0.8769	0.7761	Tidak tersedia	0.6358
RT-DETR	0.48	0.42	0.36	Tidak tersedia

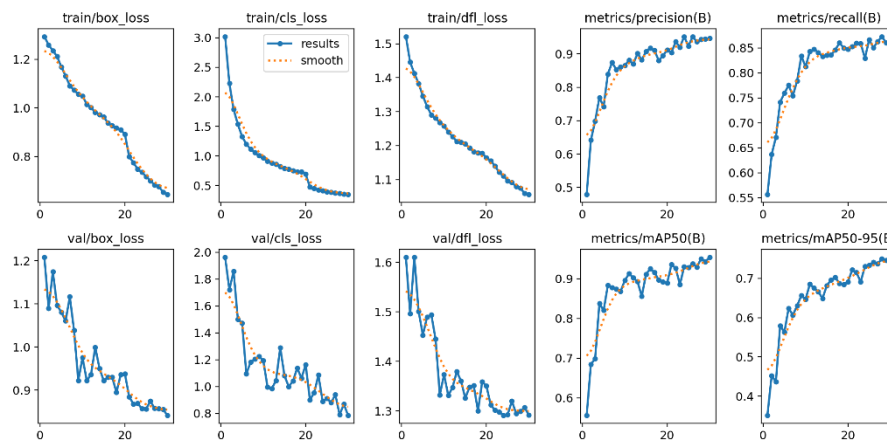
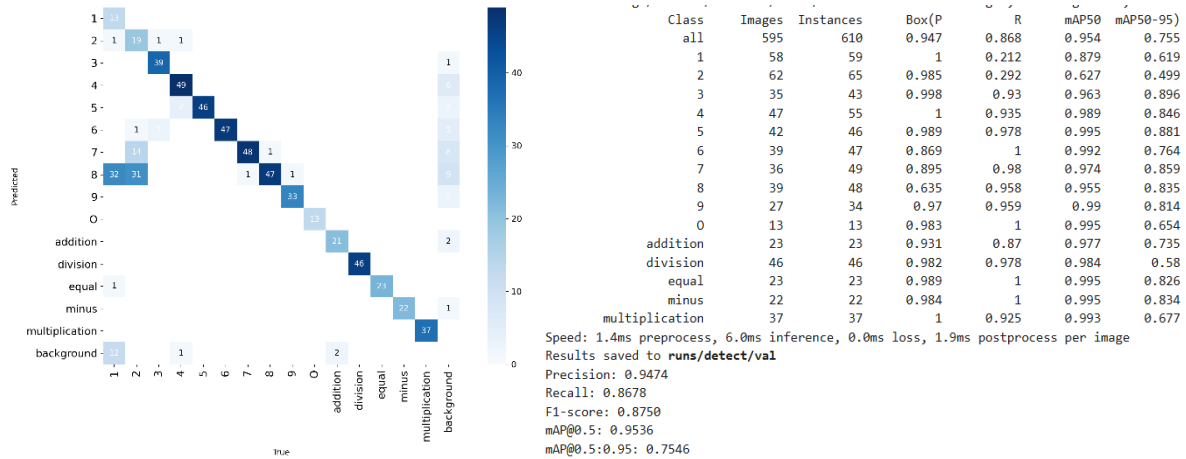
YOLOv5n mencatatkan performa terbaik di semua metrik evaluasi (mAP@0.5: 0.9474, mAP@0.75: 0.8678, mAP@0.5:0.95: 0.7546), menunjukkan akurasi deteksi objek yang sangat tinggi.

Nama : Dody Adi Sancoko (1301223071)

SSDMobileNetV3 berada di posisi kedua dengan $mAP@0.5$: 0.8769, namun $mAP@0.75$ lebih rendah. Sementara RT-DETR memiliki performa yang jauh lebih rendah ($mAP@0.5$: 0.48). Secara keseluruhan, **YOLOv5n adalah pilihan terbaik jika akurasi deteksi adalah prioritas utama.**

C. Analisis Dampak terhadap Kinerja dan Efisiensi

1. YOLOv8s



Confusion matrix ini menunjukkan bagaimana model mengklasifikasikan berbagai kategori, termasuk angka dan simbol matematika seperti addition, division, dan multiplication. **Diagonal utama menunjukkan prediksi yang benar**, seperti pada angka 1 yang berhasil diprediksi dengan benar 13 kali dan angka 9 sebanyak 49 kali. Namun, ada beberapa kesalahan, seperti angka 3 yang salah diklasifikasikan sebagai angka 4 sebanyak 39 kali. Metrik ini juga mencatat kesalahan dalam mengenali operasi matematika, dengan 21 kesalahan pada addition dan 23 kesalahan pada division. Ini memberikan gambaran area yang perlu diperbaiki untuk meningkatkan akurasi model. Model YOLOv5n menunjukkan kinerja yang sangat baik dengan **precision 0.9474**, **recall 0.8678**, dan **F1-score 0.8750**, menunjukkan keseimbangan yang baik antara akurasi dan deteksi objek. Nilai $mAP@0.5$ yang tinggi (0.9536) menandakan akurasi deteksi objek yang sangat baik pada threshold tumpang tindih 50%, meskipun $mAP@0.5:0.95$ sedikit menurun ke 0.7546, tetap menunjukkan kinerja solid pada berbagai tingkat tumpang tindih.

2. SSD MobilenetV3

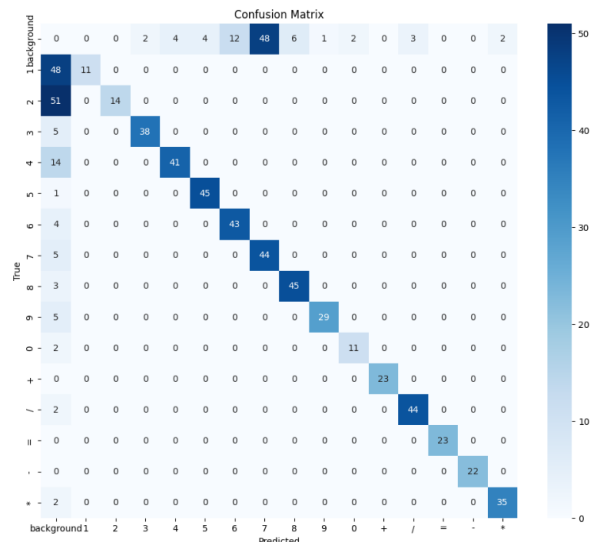
Confusion matrix ini menunjukkan kinerja model dalam mengklasifikasikan objek, dengan diagonal utama yang menunjukkan prediksi benar, seperti 48 untuk "background" dan 51 untuk "1". Meskipun ada beberapa kesalahan prediksi, **model cukup akurat dalam mendeteksi objek** relevan, meskipun

Nama : Dody Adi Sancoko (1301223071)

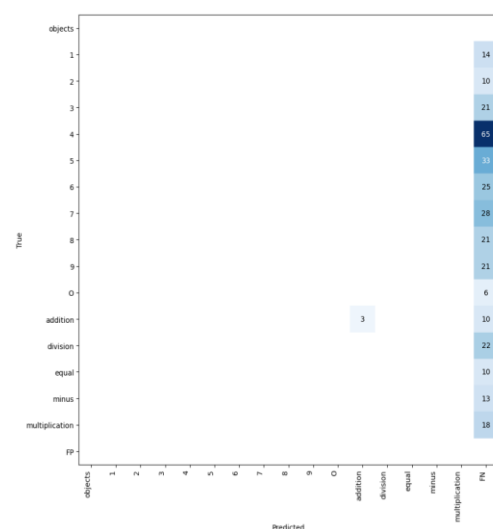
kesalahan terjadi pada beberapa kategori. Matriks ini memberi gambaran tentang area yang perlu perbaikan untuk mengurangi kesalahan klasifikasi. Model ini menunjukkan kinerja baik pada simbol matematika (addition, division, multiplication) dengan **Precision, Recall, dan F1-score hampir sempurna** (1.000). Namun, pada beberapa kelas seperti Class 1 dan background, Precision tinggi namun Recall rendah, menandakan kesulitan dalam mendeteksi objek yang relevan. Nilai mAP keseluruhan (0.6358) menunjukkan kinerja baik, dengan mAP@0.5 yang tinggi (0.8769), tetapi sedikit penurunan pada mAP@0.75 (0.7761), yang menunjukkan potensi perbaikan untuk deteksi objek pada threshold lebih ketat.

Class background: Precision=0.000, Recall=0.000, F1-score=0.000
Class 1: Precision=1.000, Recall=0.186, F1-score=0.314
Class 2: Precision=1.000, Recall=0.215, F1-score=0.354
Class 3: Precision=0.950, Recall=0.884, F1-score=0.916
Class 4: Precision=0.911, Recall=0.745, F1-score=0.820
Class 5: Precision=0.918, Recall=0.978, F1-score=0.947
Class 6: Precision=0.782, Recall=0.915, F1-score=0.843
Class 7: Precision=0.478, Recall=0.898, F1-score=0.624
Class 8: Precision=0.882, Recall=0.938, F1-score=0.909
Class 9: Precision=0.967, Recall=0.853, F1-score=0.906
Class 0: Precision=0.846, Recall=0.846, F1-score=0.846
Class +: Precision=1.000, Recall=1.000, F1-score=1.000
Class /: Precision=0.936, Recall=0.957, F1-score=0.946
Class =: Precision=1.000, Recall=1.000, F1-score=1.000
Class -: Precision=1.000, Recall=1.000, F1-score=1.000
Class *: Precision=0.946, Recall=0.946, F1-score=0.946

Overall mAP (IoU=0.5:0.95): 0.6358
mAP@0.5: 0.8769
mAP@0.75: 0.7761



3. RT-DETR



Epoch	Training Loss	Validation Loss	map	map 50	map 75	map Small	map Medium	map Large	mar 1	mar 10	mar 100
1	16.567900	21.723562	0.103700	0.163500	0.111400	0.000000	0.062100	0.128900	0.310000	0.418200	0.454700
2	12.707300	22.410326	0.067100	0.106700	0.073300	0.450000	0.039200	0.084600	0.249000	0.361000	0.366700
3	11.569700	24.101320	0.123900	0.169200	0.154000	0.150000	0.154300	0.130100	0.268500	0.319300	0.325000
4	10.733300	25.081770	0.046800	0.059300	0.059900	0.000000	0.056900	0.056500	0.157400	0.199000	0.207700
5	10.456400	25.131187	0.108400	0.139200	0.129700	0.100000	0.099700	0.109400	0.170500	0.190900	0.193600
6	10.301100	25.152550	0.063300	0.099700	0.075700	0.000000	0.076300	0.063300	0.130700	0.135100	0.135100
7	9.983700	23.924553	0.176000	0.266800	0.197700	0.000000	0.138800	0.195700	0.324100	0.352600	0.357600
8	9.894000	26.476606	0.012800	0.017500	0.014900	0.000000	0.000200	0.017800	0.055100	0.060700	0.060700
9	9.332300	26.639677	0.016500	0.022800	0.021900	0.000000	0.000000	0.021200	0.039900	0.040100	0.040100
10	9.190300	26.417210	0.004700	0.007200	0.005000	0.000000	0.000000	0.011500	0.019100	0.023600	0.023600
11	9.057100	26.239452	0.059800	0.080600	0.065400	0.000000	0.092500	0.071900	0.116900	0.139800	0.141400
12	8.831400	26.252600	0.053100	0.065900	0.064600	0.000000	0.013800	0.060200	0.104600	0.105100	0.105100
13	8.998800	27.116924	0.029800	0.046200	0.032900	0.000000	0.008900	0.033800	0.053700	0.056700	0.056700
14	8.384600	26.355894	0.084300	0.111600	0.097700	0.000000	0.093700	0.094400	0.161500	0.173900	0.173900
15	8.173500	26.289444	0.063100	0.081400	0.072600	0.000000	0.080800	0.068400	0.130200	0.135600	0.135600
16	8.172000	27.015738	0.015900	0.023600	0.018200	0.000000	0.048300	0.013400	0.026400	0.028000	0.028000
17	7.991400	27.241777	0.022000	0.038600	0.038500	0.000000	0.005300	0.043500	0.044800	0.044800	0.044800
18	7.914700	25.968166	0.132800	0.167500	0.154100	0.000000	0.107400	0.144200	0.237500	0.265000	0.265000
19	7.884700	25.426975	0.108000	0.136100	0.120100	0.250000	0.142900	0.121600	0.224100	0.257000	0.257000
20	7.473300	27.945917	0.022300	0.029500	0.025500	0.000000	0.000000	0.023700	0.025900	0.025900	0.025900
21	7.369400	26.857582	0.031700	0.042400	0.034500	0.000000	0.015500	0.022600	0.047800	0.050900	0.050900
22	7.236300	26.757021	0.070000	0.092900	0.085600	0.000000	0.008300	0.080000	0.117000	0.118100	0.118100
23	7.116300	26.772139	0.131300	0.170200	0.153600	0.000000	0.129500	0.140800	0.182300	0.211000	0.211000
24	7.039400	27.786427	0.079600	0.097600	0.090300	0.000000	0.031300	0.095000	0.088900	0.108100	0.108100
25	6.919600	26.363356	0.064900	0.079600	0.078600	0.000000	0.034900	0.071500	0.079400	0.097700	0.097700
26	6.748700	28.319006	0.062200	0.077200	0.073800	0.000000	0.065300	0.066900	0.075500	0.084800	0.084800
27	6.610100	28.372395	0.042800	0.051800	0.049000	0.000000	0.024100	0.043500	0.046000	0.054300	0.054300
28	6.535000	28.279123	0.034900	0.040100	0.040100	0.000000	0.016100	0.037400	0.035700	0.043000	0.043000
29	6.426800	28.332045	0.031700	0.035300	0.035300	0.000000	0.017500	0.036900	0.036500	0.043500	0.043500
30	6.405500	28.350779	0.036700	0.041900	0.041900	0.000000	0.016600	0.038700	0.037900	0.046300	0.046300

map50_95: 0.03
map50: 0.04
map75: 0.04

Pada 30 epoch, model menunjukkan **penurunan training loss yang stabil dan peningkatan konsisten pada mAP dan mAR**, meskipun ada fluktuasi pada validation loss. Model pada 10 epoch menunjukkan penurunan training loss yang cepat, tetapi masih fluktuatif pada validation loss dan mAP, menunjukkan ketidakstabilan. Secara keseluruhan, model lebih stabil dan optimal pada 30 epoch, meskipun perlu pelatihan lebih lanjut untuk mencapai kinerja optimal. Nilai mAP50 (0.48) menunjukkan akurasi moderat pada IoU 50%, namun penurunan pada mAP75 (0.42) dan mAP50_95 (0.36) menunjukkan penurunan kinerja pada ambang batas yang lebih ketat.