

BAB IV PENGUJIAN DAN ANALISA

Pengujian dilakukan untuk mengevaluasi pengaruh metode optimasi hiperparameter dan pembaruan dataset terhadap performa deteksi tingkat kematangan apel. Model yang dioptimasi menggunakan metode One Factor at a Time (OFAT), yaitu dengan mengubah satu hiperparameter pada setiap iterasi sementara hiperparameter lainnya dipertahankan tetap, dibandingkan dengan model hasil optimasi menggunakan metode Tree-structured Parzen Estimator (TPE). Selain itu, dilakukan pula evaluasi terhadap model setelah pembaruan dataset untuk mengetahui pengaruh penambahan variasi warna apel terhadap kemampuan model dalam melakukan klasifikasi pada kondisi nyata.

4.1 Dataset

Pada proses penyusunan dataset, dilakukan perubahan pada strategi anotasi untuk meningkatkan kemampuan model dalam mengenali variasi visual apel. Perubahan ini dilakukan karena metode anotasi pada dataset lama masih memiliki keterbatasan dalam merepresentasikan variasi warna yang muncul pada varietas apel tertentu, khususnya apel Fuji. Selain perubahan strategi anotasi, dilakukan pula pembaruan dan penambahan data untuk meningkatkan keragaman dataset. Perbandingan antara proses anotasi pada dataset lama dan dataset baru ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Anotasi kedua dataset

Pada dataset lama, proses pelabelan dilakukan berdasarkan varietas apel yang digunakan. Kedua dataset memiliki kelas yang sama, yaitu *apple-100* yang berarti apel matang, *apple-75* yang berarti apel setengah matang, dan *apple-20* yang berarti apel belum matang. Tiga varietas yang digunakan yaitu Red Delicious, Fuji, dan Granny Smith masing-masing diberi satu kelas tetap dengan tanda anotasi berwarna merah, kuning, dan hijau, yaitu *apple-100* untuk Red Delicious, *apple-75* untuk Fuji, dan *apple-20* untuk Granny Smith. Pendekatan ini mengasumsikan bahwa setiap varietas selalu mewakili satu tingkat kematangan tertentu. Padahal, khususnya pada apel Fuji, karakteristik warna dapat bervariasi meskipun berada pada tingkat kematangan yang sama. Apel Fuji dapat memiliki kombinasi warna merah, merah muda, kuning, hingga hijau, sehingga satu kelas tetap tidak mampu merepresentasikan seluruh variasi visual yang dimilikinya.

Dataset lama pada awalnya terdiri dari 500 citra. Untuk meningkatkan jumlah dan keragaman data, dilakukan proses augmentasi sehingga jumlah citra yang digunakan meningkat menjadi 1.024 citra. Meskipun demikian, keterbatasan strategi anotasi pada dataset tersebut masih menyebabkan variasi visual tertentu belum terwakili secara optimal.

Berdasarkan keterbatasan tersebut, dilakukan perubahan strategi anotasi pada dataset baru. Pelabelan tidak lagi didasarkan pada varietas apel, melainkan pada karakteristik visual warna yang menunjukkan tingkat kematangan. Dengan demikian, apel Fuji dapat diberi label *apple-20*, *apple-75*, maupun *apple-100* sesuai dengan kondisi warna pada masing-masing citra. Perubahan ini memungkinkan satu varietas apel memiliki kelas kematangan yang berbeda apabila kondisi visualnya berbeda.

Selain perubahan strategi anotasi, dataset baru juga diperbarui dengan menambahkan variasi citra dan menerapkan augmentasi untuk memperkaya karakteristik visual yang dipelajari model. Dataset baru memiliki 1.200 citra awal yang kemudian dikembangkan menjadi 3.321 citra setelah proses augmentasi.

Peningkatan jumlah data tersebut disertai dengan variasi warna, varietas, pencahayaan, sudut pengambilan, orientasi, ukuran, dan latar belakang. Dengan strategi anotasi dan keragaman data yang diperbarui, dataset baru diharapkan lebih representatif terhadap kondisi visual apel yang sebenarnya sehingga dapat meningkatkan kemampuan generalisasi model dalam mengenali tingkat kematangan pada berbagai variasi kondisi.

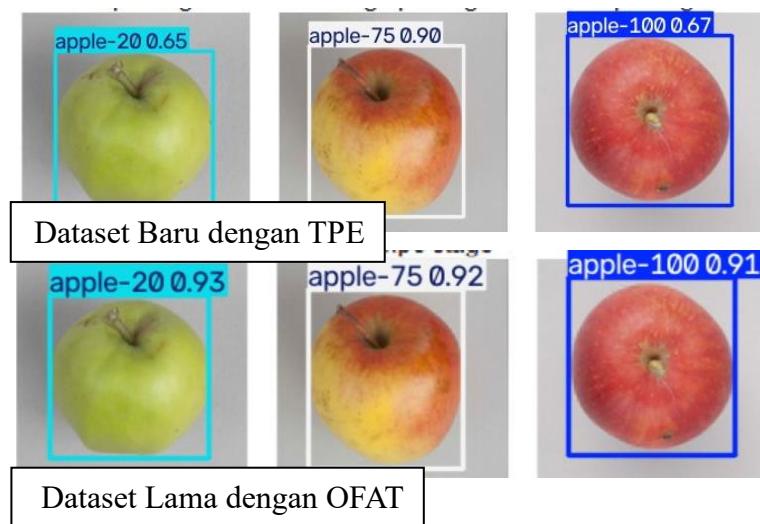
4.2 Hasil dan analisa pengujian Yolov8n

Pada pengujian pertama dilakukan perbandingan antara model OFAT yang dilatih menggunakan dataset lama dengan model yang dilatih menggunakan dataset baru. Fokus pengujian adalah kemampuan model dalam mendeteksi apel Malang yang secara alami memiliki warna hijau kekuningan atau hijau kemerahan meskipun telah matang.

Berdasarkan hasil deteksi, model yang dilatih menggunakan dataset lama masih mengalami kesalahan klasifikasi terhadap apel Malang matang. Objek tersebut tidak dikenali sebagai kelas apple-100 karena dataset lama didominasi oleh sampel apel matang yang berwarna merah cerah, sehingga model mempelajari bahwa karakteristik utama apel matang adalah warna merah. Akibatnya, apel matang dengan variasi warna hijau atau kekuningan sering diklasifikasikan sebagai kelas lain.

Setelah dataset diperbarui dengan menambahkan berbagai variasi warna apel matang, model menunjukkan peningkatan kemampuan generalisasi. Pada hasil deteksi menggunakan dataset baru, apel Malang yang memiliki warna hijau kemerahan maupun kekuningan berhasil dikenali sebagai kelas apple-100 sesuai kondisi sebenarnya. Hasil ini menunjukkan bahwa peningkatan keragaman data pelatihan memberikan representasi fitur yang lebih baik sehingga model tidak lagi bergantung hanya pada warna merah sebagai indikator utama tingkat kematangan. Dengan demikian, model menjadi lebih robust terhadap variasi karakteristik visual yang ditemukan pada kondisi nyata. Hal ini bisa divisualisasikan pada gambar 4.2 berikut:

Perbedaan nilai *confidence score* tersebut menunjukkan bahwa masing-masing model memiliki tingkat keyakinan yang berbeda dalam melakukan prediksi terhadap objek yang sama. Semakin tinggi nilai *confidence score*, semakin tinggi pula tingkat keyakinan model bahwa objek yang dideteksi termasuk ke dalam kelas yang diprediksi. Oleh karena itu, perbedaan nilai ini mencerminkan perbedaan karakteristik model sebagai hasil dari metode optimasi hiperparameter yang digunakan, yang juga konsisten dengan perbedaan performa yang ditunjukkan pada hasil pengujian sebelumnya. Visualisasi hasil perbandingan deteksi kedua model dapat dilihat pada Gambar 4.2 dan gambar 4.3.



Gambar 4. 3 Hasil Deteksi TPE dan OFAT

Pengujian terakhir membandingkan performa model OFAT dan TPE yang sama-sama dilatih menggunakan dataset baru. Perbandingan difokuskan pada kemampuan kedua model dalam mendeteksi objek berdasarkan karakteristik precision, recall, dan F1-score. Pengujian ini dilakukan menggunakan satu gambar yang memiliki 14 objek apel sebagai ground truth. Oleh karena itu, hasil pengujian pada tahap ini digunakan sebagai demonstrasi atau contoh untuk menggambarkan perbedaan karakteristik deteksi antara kedua model dan belum dapat dianggap sebagai representasi performa keseluruhan.

Berdasarkan hasil pengujian, model TPE menghasilkan nilai precision, recall, dan F1-score yang lebih tinggi dibandingkan model OFAT. Model TPE memperoleh

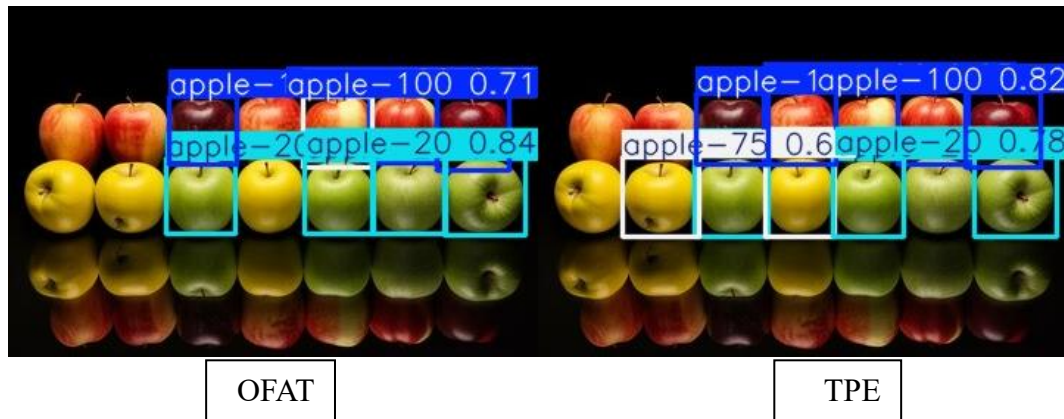
nilai precision sebesar 1,000, yang menunjukkan bahwa seluruh prediksi yang dihasilkan pada gambar pengujian merupakan prediksi yang benar dan tidak terdapat false positive. Selain itu, model TPE memperoleh nilai recall sebesar 0,714, yang menunjukkan bahwa model mampu mendeteksi 10 dari 14 objek yang terdapat pada gambar. Dengan demikian, masih terdapat 4 objek yang tidak berhasil dideteksi atau termasuk dalam kategori false negative.

Sementara itu, model OFAT memperoleh nilai precision sebesar 0,889, recall sebesar 0,615, dan F1-score sebesar 0,727. Nilai tersebut diperoleh dari 8 true positive, 1 false positive, dan 5 false negative. Model OFAT menghasilkan jumlah prediksi yang lebih sedikit dibandingkan model TPE, yaitu sebanyak 9 prediksi. Dari jumlah tersebut, 8 prediksi merupakan deteksi yang benar, sedangkan 1 prediksi merupakan deteksi yang tidak sesuai dengan ground truth. Model OFAT juga tidak berhasil mendeteksi 5 objek yang terdapat pada gambar pengujian.

Hasil tersebut menunjukkan bahwa pada gambar pengujian ini, model TPE memiliki kemampuan deteksi yang lebih baik dibandingkan model OFAT. Model TPE tidak hanya menghasilkan precision yang sempurna, tetapi juga mampu mendeteksi lebih banyak objek sehingga memperoleh nilai recall yang lebih tinggi. Kombinasi nilai precision dan recall tersebut menghasilkan F1-score sebesar 0,833, sedangkan model OFAT memperoleh F1-score sebesar 0,727. Dengan demikian, berdasarkan hasil pengujian pada gambar ini, model TPE menunjukkan keseimbangan antara ketepatan prediksi dan kemampuan menemukan objek yang lebih baik dibandingkan model OFAT.

Perbedaan hasil deteksi dapat dipengaruhi oleh karakteristik dataset baru yang memiliki variasi visual lebih tinggi dibandingkan dataset sebelumnya. Dataset tersebut mencakup beberapa varietas apel dengan karakteristik warna yang berbeda, seperti apel Granny Smith yang tetap berwarna hijau meskipun telah matang. Selain itu, terdapat variasi kondisi pencahayaan, sudut pengambilan gambar, orientasi buah, ukuran objek, serta perbedaan latar belakang. Variasi tersebut menyebabkan distribusi fitur antar kelas menjadi lebih kompleks sehingga model dapat

menghasilkan tingkat keyakinan atau confidence score yang berbeda terhadap objek dengan karakteristik visual yang serupa.



Gambar 4. 4 Hasil Deteksi OFAT dan TPE pada dataset baru

Pada tabel yang nanti akan disajikan, akan menunjukkan hasil perhitungan bounding box yang menjadi dasar dalam evaluasi performa deteksi kedua model. Pada model OFAT, dari 14 objek ground truth dihasilkan 9 prediksi yang terdiri atas 8 true positive (TP), 1 false positive (FP), dan 5 false negative (FN). Sementara itu, model TPE menghasilkan 10 prediksi yang seluruhnya merupakan true positive tanpa menghasilkan false positive, namun masih terdapat 4 objek yang tidak berhasil dideteksi. Berdasarkan hasil tersebut, model TPE mampu mendeteksi lebih banyak objek dibandingkan model OFAT sekaligus menghasilkan prediksi yang lebih tepat pada gambar pengujian.

Nilai precision, recall, dan F1-score pada Tabel 4.2 dihitung berdasarkan jumlah true positive, false positive, dan false negative yang diperoleh pada Tabel 4.1. Model OFAT memperoleh nilai precision sebesar 0,889, recall sebesar 0,615, dan F1-score sebesar 0,727. Sementara itu, model TPE memperoleh nilai precision sebesar 1,000, recall sebesar 0,714, dan F1-score sebesar 0,833. Nilai precision sebesar 1,000 pada model TPE menunjukkan bahwa seluruh objek yang diprediksi oleh model merupakan deteksi yang benar. Selain itu, nilai recall yang lebih tinggi menunjukkan bahwa model TPE mampu menemukan lebih banyak objek yang terdapat pada gambar dibandingkan model OFAT. Oleh karena itu, berdasarkan

hasil pengujian pada gambar tersebut, model TPE menghasilkan performa deteksi yang lebih baik dan lebih seimbang dibandingkan model OFAT. berikut tabel 4.1:

Tabel 4. 1 Perhitungan Bounding Box

Gambar	Ground Truth Count	Prediction	TP	FP	FN
OFAT	14	9	8	1	5
TPE	14	10	10	0	4

Namun, hasil pengujian ini hanya diperoleh dari satu gambar sehingga jumlah sampel yang digunakan masih terbatas. Oleh karena itu, nilai precision, recall, dan F1-score yang diperoleh belum dapat menggambarkan performa kedua model secara keseluruhan dengan tingkat akurasi yang tinggi. Pengujian ini digunakan sebagai demonstrasi atau contoh untuk memperlihatkan perbedaan hasil deteksi antara model OFAT dan TPE pada kondisi gambar tertentu. Evaluasi yang lebih komprehensif akan dilakukan melalui pengujian sistem pada perangkat embedded menggunakan Raspberry Pi Camera untuk melakukan deteksi secara real-time. Pengujian tersebut akan menggunakan lebih banyak objek apel dan dilakukan pada bagian pengujian throughput. Hasil pengujian throughput diharapkan dapat memberikan gambaran yang lebih representatif mengenai kemampuan model dalam mendeteksi dan melakukan proses grading apel pada kondisi operasional sistem yang sebenarnya.:

Tabel 4. 2 Performa Deteksi

Metode	Precision	Recall	F1-Score
(OFAT)	0.889	0.615	0.727
(TPE)	1.000	0.714	0.833

4.3 Hasil dan Analisa Metode TPE (Optuna)

Optimasi hyperparameter pada penelitian ini dilakukan menggunakan framework Optuna dengan algoritma Tree-Structured Parzen Estimator (TPE). Hyperparameter yang dioptimasi meliputi learning rate (lr0), weight decay, optimizer, dan batch size. Rentang pencarian ditetapkan pada learning rate 0,001–0,1, weight decay 0,00005–0,05, optimizer SGD, Adam, dan AdamW, serta batch size 8, 16, 32, dan 64. Fungsi objektif (*fitness*) dibentuk menggunakan kombinasi 80% F1-Score dan 20% mAP@50, sehingga proses optimasi lebih menekankan keseimbangan antara precision dan recall tanpa mengabaikan kualitas deteksi secara keseluruhan. Sebanyak 50 trial dilakukan, di mana setiap trial melatih model selama 40 epoch. Berdasarkan log Optuna, kombinasi hiperparameter optimal tercapai pada tabel di bawah

Tabel 4. 3 Hasil 50 Trial TPE

	Fitness	BS	Learning Rate	Opti- mizer	Weight Decay	F1	mAP 50
0	0.856189	32	0.005612	SGD	0.035572	0.848289	0.887788
1	0.824563	8	0.015931	Adam	0.006656	0.813871	0.867334
2	0.852487	32	0.00406	Adam W	0.001876	0.845234	0.881499
3	0.854139	16	0.008168	Adam W	0.011337	0.8462	0.885893
4	0.84143	16	0.001349	SGD	0.035126	0.835717	0.864281
5	0.840166	8	0.001754	Adam	0.001529	0.834328	0.863516
6	0.859163	32	0.0124	SGD	0.000179	0.856009	0.871776
7	0.845008	16	0.001503	Adam W	0.000194	0.840153	0.864427
8	0.83892	8	0.003646	Adam	0.002124	0.82867	0.879918
9	0.842596	16	0.001026	Adam W	0.013975	0.836451	0.867176
10	0.858381	16	0.053236	SGD	0.003706	0.849815	0.892641
11	0.8338	16	0.059487	Adam W	0.001305	0.83100	0.84500

12	0.837088	8	0.011104	Adam	0.000959	0.827267	0.876368
13	0.823732	16	0.065333	Adam	0.00028	0.812958	0.866827
14	0.579154	16	0.072343	Adam	0.013284	0.574693	0.596997
15	0.852973	32	0.017654	SGD	9.39E-05	0.850958	0.861031
16	0.850544	16	0.008873	SGD	0.00105	0.846953	0.864909
17	0.862791	32	0.045351	SGD	0.001332	0.856194	0.889181
18	0.844615	32	0.063239	SGD	0.00226	0.83385	0.887674
19	0.837126	32	0.002766	SGD	0.000277	0.833554	0.85141
20	0.853513	8	0.026566	SGD	0.000269	0.846273	0.882475
21	0.84478	16	0.096762	SGD	0.003505	0.834441	0.886134
22	0.859527	32	0.010345	SGD	0.001023	0.859084	0.861297
23	0.856294	32	0.008666	SGD	0.001976	0.853881	0.865943
24	0.831235	32	0.030903	Adam	0.002079	0.824044	0.859996
25	0.851393	32	0.019193	SGD	0.000519	0.846666	0.870298
26	0.85722	32	0.005823	Adam	8.43E-05	0.850304	0.884881
27	0.8464	32	0.069854	Adam W	0.00031	0.84400	0.85600
28	0.841518	8	0.004374	SGD	0.000174	0.833873	0.872101
29	0.8398	32	0.059073	Adam W	0.006705	0.83700	0.85100
30	0.844807	8	0.010376	Adam W	5.20E-05	0.835321	0.882751
31	0.853331	8	0.01509	SGD	0.00328	0.84646	0.880816
32	0.814913	8	0.075805	Adam	0.002552	0.804182	0.857839
33	0.855291	16	0.044388	SGD	0.000734	0.847081	0.888131
34	0.856915	16	0.015183	SGD	0.01216	0.84827	0.891497
35	0.857068	32	0.075061	SGD	0.000323	0.849353	0.88793
36	0.838851	32	0.012634	Adam W	0.000165	0.831704	0.86744
37	0.811844	8	0.081131	SGD	0.022389	0.802517	0.849151
38	0.851478	16	0.008046	SGD	5.90E-05	0.847923	0.865699
39	0.838183	32	0.005035	Adam	0.039115	0.830008	0.870884
40	0.863758	32	0.021759	SGD	0.002654	0.858016	0.886726

41	0.853119	32	0.017098	SGD	0.001399	0.850567	0.863329
42	0.847229	32	0.020701	SGD	0.010299	0.838694	0.881372
43	0.849494	16	0.036714	SGD	0.005586	0.84072	0.884589
44	0.849648	32	0.009671	SGD	0.00033	0.847435	0.858502
45	0.842586	32	0.0011	Adam	0.003573	0.837816	0.861666
46	0.851788	32	0.030765	SGD	0.002349	0.845837	0.875594
47	0.854891	8	0.062935	SGD	0.002136	0.846025	0.890355
48	0.848302	32	0.005518	Adam	0.00138	0.838829	0.886192
49	0.846595	32	0.009305	SGD	0.000121	0.843056	0.860752

. Hasil seluruh proses optimasi ditunjukkan pada Tabel 3.10. Berdasarkan nilai fitness, Trial 40 menghasilkan konfigurasi hyperparameter terbaik dengan nilai fitness sebesar 0,863758. Konfigurasi tersebut terdiri dari batch size 32, learning rate 0,021759, optimizer SGD, dan weight decay 0,002654, menghasilkan F1-score sebesar 0,858016 serta mAP@50 sebesar 0,886726.

Distribusi hasil trial menunjukkan bahwa sebagian besar konfigurasi terbaik menggunakan optimizer SGD. Dari sepuluh trial dengan nilai fitness tertinggi, hampir seluruhnya menggunakan SGD, sedangkan optimizer Adam hanya muncul pada beberapa konfigurasi dengan performa menengah dan AdamW tidak berhasil menempati kelompok trial terbaik. Hal ini mengindikasikan bahwa untuk dataset yang digunakan pada penelitian ini, SGD memberikan proses konvergensi yang lebih stabil dan menghasilkan keseimbangan precision serta recall yang lebih baik dibandingkan optimizer lainnya.

Selain optimizer, batch size juga menunjukkan pengaruh yang cukup signifikan terhadap performa model. Sebagian besar trial terbaik menggunakan batch size 32, sedangkan batch size 8 dan 16 menghasilkan performa yang lebih bervariasi. sehingga dapat disimpulkan bahwa batch size 32 memberikan kompromi terbaik antara stabilitas pembaruan gradien, penggunaan memori, dan kemampuan generalisasi model pada dataset ini.

Pengaruh learning rate terlihat lebih dominan dibandingkan weight decay. Trial-trial dengan performa terbaik umumnya memiliki learning rate pada kisaran 0,01 hingga 0,05, sedangkan beberapa konfigurasi dengan learning rate yang terlalu tinggi menghasilkan penurunan performa yang cukup drastis. Sebagai contoh, Trial 14 dengan learning rate 0,072343 hanya memperoleh fitness sebesar 0,579154, jauh di bawah sebagian besar trial lainnya. Kondisi tersebut menunjukkan bahwa kombinasi hyperparameter tertentu menyebabkan proses pelatihan tidak mampu menghasilkan model yang layak dievaluasi, yang merupakan konsekuensi umum pada proses eksplorasi ruang pencarian hyperparameter yang luas.

Sebaliknya, pengaruh weight decay cenderung lebih moderat. Nilai weight decay pada konfigurasi terbaik tersebar mulai dari 9.39×10^{-5} hingga sekitar 0.035, menunjukkan bahwa model masih mampu mencapai performa tinggi pada beberapa tingkat regularisasi yang berbeda selama didukung oleh kombinasi learning rate, optimizer, dan batch size yang sesuai. Hasil ini mengindikasikan bahwa interaksi antarhyperparameter memiliki pengaruh yang lebih besar dibandingkan optimasi masing-masing parameter secara independen.

Karakteristik tersebut menunjukkan keunggulan pendekatan Tree-Structured Parzen Estimator dibandingkan metode pencarian konvensional seperti OFAT. TPE tidak mengevaluasi setiap kombinasi secara acak maupun mengubah satu parameter pada satu waktu, melainkan memanfaatkan hasil trial sebelumnya untuk membangun model probabilistik mengenai area ruang pencarian yang berpotensi menghasilkan performa tinggi. Seiring bertambahnya jumlah trial, algoritma secara bertahap mengarahkan eksplorasi menuju konfigurasi yang semakin menjanjikan sehingga proporsi trial dengan performa tinggi meningkat. Pendekatan ini memungkinkan eksplorasi ruang hyperparameter yang jauh lebih efisien, terutama ketika terdapat interaksi kompleks antara learning rate, optimizer, weight decay, dan batch size.

4.5 Perbandingan Performa OFAT dan TPE

Perbandingan hasil tuning menggunakan metode One Factor at a Time (OFAT) dan Tree-structured Parzen Estimator (TPE) menunjukkan bahwa optimasi

TPE mampu memperoleh performa model yang sebanding dengan OFAT dengan waktu tuning yang jauh lebih singkat. Pengujian awal menggunakan dataset lama dilakukan pada kondisi dan konfigurasi hyperparameter yang setara, sehingga hasil tersebut dapat digunakan untuk mengevaluasi efektivitas metode optimasi yang digunakan.

Berdasarkan hasil tuning pada dataset lama, OFAT membutuhkan waktu lebih dari 8 jam, sedangkan TPE hanya membutuhkan 2,6 jam. Dengan demikian, TPE membutuhkan waktu tuning kurang dari seperlima waktu yang diperlukan OFAT, atau setidaknya sekitar 5 kali lebih cepat. Jika dibandingkan berdasarkan pengurangan waktu, penggunaan TPE menghasilkan penghematan waktu tuning minimal sebesar 80%. Perbedaan ini menunjukkan bahwa keunggulan TPE tidak hanya terletak pada kemampuan memperoleh hyperparameter yang baik, tetapi juga pada efisiensi proses pencarian hyperparameter.

Tabel 4. 4 Tabel Komparasi Dataset Lama

Aspek	OFAT	TPE
Metode	Mengubah satu hyperparameter tanpa feedback	Bayesian Optimization otomatis
Dataset	1,342	1,342
Epoch , Trial	100 , 14	30 , 30
Tuning Total Time	8 Hours	2.6 Hours
Objektif Fitness	Analisis Manual Setiap Iterasi	Diperoleh berdasarkan model probablistik

Meskipun waktu tuning TPE jauh lebih singkat, performa model yang dihasilkan tidak mengalami penurunan yang berarti. Pada tabel 3.5 dan tabel 3.6 mcode OFAT memperoleh precision sebesar 0,982, recall 0,998, F1-score 0,990,

mAP@50 sebesar 0,991, dan mAP@50–95 sebesar 0,850. Sementara itu, model TPE memperoleh precision sebesar 0,990, recall 0,990, F1-score 0,995, mAP@50 sebesar 0,995, dan mAP@50–95 sebesar 0,86523.

Pada precision, TPE menghasilkan peningkatan sebesar 0,008 atau sekitar 0,81% dibandingkan OFAT. Recall TPE memang lebih rendah sebesar 0,008 atau sekitar 0,80%, tetapi penurunan tersebut diimbangi oleh peningkatan F1-score sebesar 0,005 atau sekitar 0,51%. Pada mAP@50, TPE juga memberikan peningkatan sebesar 0,004 atau sekitar 0,40%. Perbedaan paling besar terlihat pada mAP@50–95, yaitu TPE mencapai 0,86523 dibandingkan 0,850 pada OFAT, atau meningkat sebesar 0,01523 atau sekitar 1,79%.

Berdasarkan hasil evaluasi pada dataset lama, TPE kemudian diterapkan pada dataset baru yang memiliki jumlah data lebih besar, yaitu 3.344 citra. Jumlah tersebut sekitar 2,49 kali lebih banyak dibandingkan dataset sebelumnya yang terdiri dari 1.342 citra. Pada dataset baru, proses tuning dilakukan menggunakan 50 trial dengan masing-masing 40 epoch. Peningkatan ukuran dataset dan jumlah trial menyebabkan kebutuhan komputasi menjadi lebih besar, sehingga efisiensi metode optimasi menjadi semakin penting.

Tabel 4. 4 Tabel Tuning Dataset baru

Aspek	TPE
Metode	Bayesian Optimization otomatis
Dataset	3,344
Epoch , Trial	40, 50
Tuning Total Time	9.9 Hours
Objektif Fitness	Analisis Manual Setiap Iterasi

Penggunaan 40 epoch pada setiap trial merupakan salah satu strategi untuk menjaga proses tuning tetap efisien. Tujuan utama hyperparameter optimization adalah menemukan kombinasi hyperparameter yang paling menjanjikan, bukan menghasilkan model final pada setiap percobaan. Setelah kombinasi terbaik ditemukan, hyperparameter tersebut dapat digunakan untuk pelatihan akhir dengan jumlah epoch yang lebih besar. Dengan TPE, hasil trial sebelumnya juga digunakan untuk mengarahkan pencarian ke kombinasi yang lebih menjanjikan. Pada eksperimen ini, 50 trial dengan 40 epoch menghasilkan total 2.000 epoch pelatihan. Sebagai perbandingan, apabila OFAT digunakan dengan 50 kombinasi hyperparameter dan masing-masing dilatih selama 100 epoch, diperlukan total 5.000 epoch pelatihan, atau 2,5 kali lebih banyak. Dengan dataset sebesar 3.344 citra, perbedaan jumlah pelatihan tersebut dapat menghasilkan kebutuhan waktu komputasi yang sangat besar.

Tabel 4. 5 Tabel Komparasi Retrain 50 Epoch

Aspek	OFAT	TPE
Metode	Mengubah satu hyperparameter tanpa feedback	Bayesian Optimization otomatis
Dataset	3.344	3.344
Epoch	50	50
Training Time	26 menit	27 menit
Performance Metrics	F1: 0,805 mAP50: 0,845 mAP50-95: 0,710	F1: 0,845 mAP50: 0,875 mAP50-95: 0,747

Berdasarkan Tabel 4.5, pengujian ulang (*retraining*) 50 *epoch* menegaskan superioritas *Tree-structured Parzen Estimator* (TPE) dibandingkan *One Factor at a Time* (OFAT) pada dataset baru. Dengan kondisi pelatihan yang identik, model hasil optimasi TPE secara konsisten mengungguli OFAT pada seluruh metrik performa: *F1-score* meningkat dari 0,805 menjadi 0,845, *mAP@50* dari 0,845 menjadi 0,875, dan *mAP@50-95* dari 0,710 menjadi 0,747. Waktu pelatihan akhir kedua metode terbukti ekuivalen, yakni 26 menit untuk OFAT dan 27 menit untuk TPE. Fakta ini mengonfirmasi bahwa pencarian *hyperparameter* berbasis probabilitas pada TPE menghasilkan konfigurasi yang secara langsung meningkatkan efektivitas ekstraksi fitur, tanpa menambah beban komputasi pada fase pelatihan akhir.

Superioritas performa model tersebut berbanding lurus dengan alokasi waktu *tuning* TPE yang berdurasi 9,9 jam. Peningkatan durasi *tuning* ini proporsional terhadap ekspansi volume dataset dari 1.342 menjadi 3.344 citra. Eksperimen ini secara konklusif membuktikan bahwa seiring dengan peningkatan dimensi dataset dan kompleksitas ruang *hyperparameter*, implementasi optimasi adaptif seperti TPE merupakan metode yang esensial. Eksplorasi terarah pada fase *tuning* menjamin perolehan kombinasi *hyperparameter* optimal yang memaksimalkan metrik evaluasi model akhir secara terukur dan efisien.

Proses *tuning* TPE pada dataset baru membutuhkan waktu total 9,9 jam. Meskipun waktu ini lebih panjang dibandingkan *tuning* TPE pada dataset lama yang hanya membutuhkan 2,6 jam, kondisi eksperimennya juga lebih berat karena dataset meningkat dari 1.342 menjadi 3.344 citra. Hasil pada dataset lama telah menunjukkan bahwa TPE mampu memperoleh performa yang hampir sama bahkan sedikit lebih baik dibandingkan OFAT dengan waktu *tuning* yang jauh lebih singkat. Oleh karena itu, penggunaan TPE pada dataset baru dipilih untuk menghindari kebutuhan komputasi yang jauh lebih besar apabila seluruh kombinasi *hyperparameter* harus diuji menggunakan pendekatan OFAT dengan *epoch* yang panjang.

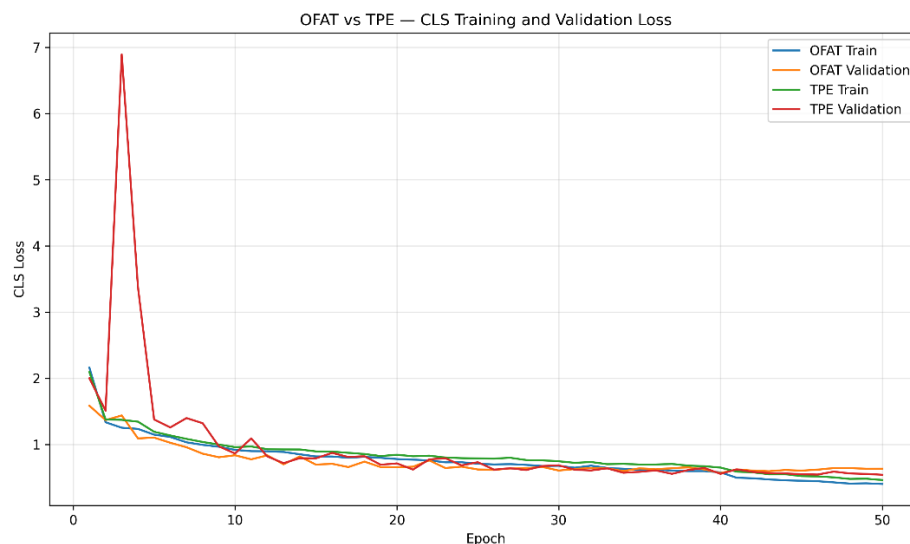
Dengan demikian, penerapan TPE pada dataset baru tidak hanya bertujuan untuk memperoleh nilai metrik yang optimal, tetapi juga untuk menjaga efisiensi proses pencarian hyperparameter. Hasil eksperimen menunjukkan bahwa ketika ukuran dataset dan jumlah percobaan semakin besar, penggunaan metode optimasi adaptif seperti TPE menjadi semakin relevan karena pencarian dapat dilakukan dengan jumlah epoch yang lebih rendah pada tahap tuning, kemudian kombinasi hyperparameter terbaik digunakan untuk pelatihan akhir. Pendekatan ini memungkinkan ruang hyperparameter dieksplorasi secara lebih luas tanpa menyebabkan kebutuhan waktu komputasi meningkat secara tidak terkendali.

4.6 Analisis Efisiensi Optimasi

Pada ada penelitian sebelumnya, perbandingan metode *One Factor at a Time* (OFAT) dan *Tree-Structured Parzen Estimator* (TPE) menunjukkan bahwa TPE mampu melakukan pencarian *hyperparameter* secara lebih adaptif dengan memanfaatkan informasi dari percobaan sebelumnya. Pendekatan tersebut memungkinkan pencarian konfigurasi dilakukan secara lebih terarah dibandingkan OFAT yang mengubah satu *hyperparameter* pada setiap percobaan.

Berdasarkan pertimbangan tersebut, penelitian ini menggunakan TPE dengan sumber daya komputasi yang lebih besar. Penelitian sebelumnya menggunakan 30 *trial* dengan 30 *epoch* dan membutuhkan waktu sekitar 2,6 jam, sedangkan penelitian ini menggunakan dataset yang lebih besar sehingga optimasi diperluas menjadi 50 *trial* dengan 40 *epoch* pada setiap *trial*. Proses tersebut membutuhkan waktu sekitar 599 menit atau hampir 10 jam. Waktu tersebut menunjukkan besarnya kebutuhan komputasi ketika jumlah data, *trial*, dan *epoch* ditingkatkan. Apabila pencarian dengan jumlah percobaan yang setara dilakukan menggunakan OFAT secara ekstensif, kebutuhan komputasinya akan meningkat secara signifikan karena OFAT tidak memanfaatkan hubungan antarkombinasi *hyperparameter* dan harus mengevaluasi konfigurasi secara lebih terpisah. Dengan demikian, penggunaan OFAT pada ruang pencarian yang besar seperti pada penelitian ini berpotensi membutuhkan waktu dan sumber daya komputasi yang jauh lebih besar sehingga menjadi kurang praktis.

Hasil optimasi kemudian digunakan untuk melakukan *retraining* model OFAT dan TPE selama 50 epoch. Kurva *training loss* dan *validation loss* ditunjukkan pada Gambar 4.5.



Gambar 4. 5 Gambar Analisa kurva CLS Loss

Pada kedua model, *training loss* dan *validation loss* secara umum mengalami penurunan hingga akhir pelatihan. Meskipun terdapat fluktuasi pada *validation loss*, tidak terlihat pola peningkatan *validation loss* yang berlangsung terus-menerus ketika *training loss* tetap menurun. Kondisi tersebut menunjukkan bahwa 50 epoch telah menghasilkan proses pelatihan yang relatif stabil dan belum menunjukkan indikasi *overfitting* yang kuat. Oleh karena itu, 50 epoch dapat dianggap telah mencukupi untuk proses *retraining* pada dataset apel penelitian ini. Penggunaan 100, 150, atau 200 epoch seperti pada penelitian Rifatatiar [4] tidak menjadi kebutuhan pada dataset yang digunakan dalam penelitian ini.

Hasil evaluasi menunjukkan bahwa model hasil optimasi TPE memperoleh nilai *precision*, *mAP@50*, dan *mAP@50–95* yang lebih tinggi pada evaluasi *best.pt*, serta mencapai nilai *peak* yang lebih baik pada sebagian besar metrik dibandingkan model OFAT. Hasil tersebut menunjukkan bahwa tambahan sumber daya komputasi yang dialokasikan pada proses optimasi TPE dapat menghasilkan konfigurasi *hyperparameter* yang lebih optimal.

Optimasi TPE juga tidak memberikan beban komputasi tambahan yang berarti pada tahap inferensi. Model TPE memiliki waktu *preprocessing* 2,15 ms, inferensi 6,86 ms, dan *postprocessing* 1,18 ms, dengan total waktu pemrosesan 10,19 ms per citra atau 98,16 FPS. Sementara itu, model OFAT membutuhkan total waktu 10,27 ms per citra atau 97,37 FPS. Selisih sekitar 0,08 ms menunjukkan bahwa optimasi *hyperparameter* tidak meningkatkan kompleksitas inferensi secara berarti karena kedua model tetap menggunakan arsitektur YOLOv8 yang sama.

Secara keseluruhan, TPE memungkinkan pencarian *hyperparameter* yang lebih adaptif dengan konsekuensi waktu optimasi yang cukup besar, yaitu sekitar 599 menit pada penelitian ini. Namun, peningkatan sumber daya tersebut menghasilkan konfigurasi dengan performa deteksi yang lebih tinggi tanpa menambah kompleksitas pada tahap inferensi. Selain itu, hasil kurva *training loss* dan *validation loss* menunjukkan bahwa 50 *epoch* telah memberikan kondisi pelatihan yang relatif stabil pada dataset apel yang digunakan. Dengan demikian, peningkatan jumlah *epoch* hingga 100–200 *epoch* tidak diperlukan dalam proses *retraining* pada penelitian ini, sedangkan penggunaan OFAT secara ekstensif pada ruang pencarian yang sama berpotensi membutuhkan waktu komputasi yang jauh lebih besar.

4.7 Throughput Sistem

Pada penelitian sebelumnya, apel diletakkan secara manual ke atas konveyor oleh operator sehingga jarak antar apel dan frekuensi pemasukan sangat bergantung pada konsistensi manusia. Meskipun kecepatan konveyor telah ditentukan, laju masuk apel ke dalam sistem masih dapat bervariasi akibat perbedaan waktu penempatan setiap apel. Kondisi tersebut berpotensi memengaruhi hasil pengujian throughput karena aliran objek yang diproses tidak selalu seragam.

Pada penelitian ini, motor DC 12 V digunakan sebagai mekanisme *feeder* untuk mengatur suplai apel menuju konveyor utama. Mekanisme *feeder* menggunakan roda silikon yang menyalurkan apel melalui saluran dengan kemiringan 10° menuju konveyor utama. Kecepatan putaran motor diatur menggunakan *DC dimmer* melalui proses kalibrasi hingga diperoleh laju

pemasukan apel yang mendekati 36 apel per menit. Dimmer diatur untuk mendapatkan 3.5 V dan menghasilkan 30 rpm.

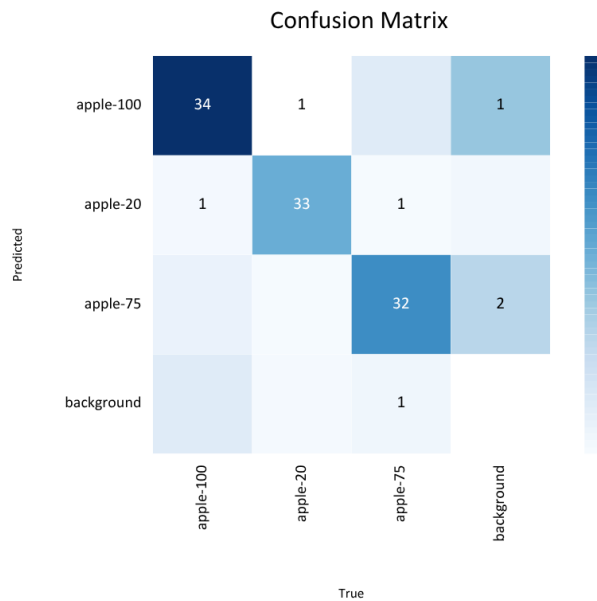
Berdasarkan hasil pengujian pada Tabel 4.5 yang akan disajikan di bawah, jumlah apel yang diproses dalam setiap pengujian tidak selalu sama, yaitu sebanyak 34 apel, 36 apel, dan 33 apel per menit. Perbedaan jumlah apel tersebut dipengaruhi oleh variasi ukuran dan massa apel yang digunakan pada setiap pengujian. Apel dengan ukuran atau massa yang lebih besar dapat menghasilkan beban mekanis yang lebih tinggi pada roda silikon dan motor *feeder*. Peningkatan beban tersebut dapat menyebabkan kecepatan putaran motor sedikit menurun sehingga jumlah apel yang disalurkan menuju konveyor dalam waktu satu menit menjadi lebih sedikit. Sebaliknya, ketika apel yang digunakan memiliki ukuran atau massa yang lebih kecil, beban yang diterima motor menjadi lebih rendah sehingga mekanisme *feeder* dapat menyalurkan apel dengan laju yang lebih tinggi. Oleh karena itu, meskipun nilai pengaturan pada *DC dimmer* dibuat tetap, jumlah apel yang masuk ke sistem dapat mengalami variasi akibat perbedaan karakteristik fisik apel. Data akan disajikan pada tabel di bawah:

Tabel 4.5 *Keluaran Apel Permenit*

Jumlah Apel	TP	FP	FN	Precision	Recall	F1
34	33	1	0	0,971	1,000	0,985
36	34	2	0	0,944	1,000	0,971
33	32	0	1	1,000	0,970	0,985

Pada pengujian pertama, sistem memproses 34 apel dalam satu menit dengan 33 *true positive* (TP), 1 *false positive* (FP), dan tanpa *false negative* (FN). Hasil tersebut menghasilkan nilai *precision* sebesar 0,971, *recall* sebesar 1,000, dan F1-score sebesar 0,985. Pada pengujian kedua, sistem memproses 36 apel dengan 34 TP, 2 FP, dan tanpa FN. Nilai *precision*, *recall*, dan F1-score yang diperoleh masing-

masing sebesar 0,944, 1,000, dan 0,971. Sementara itu, pada pengujian ketiga, sistem memproses 33 apel dengan 32 TP, tanpa FP, dan 1 FN. Hasil tersebut menghasilkan nilai *precision* sebesar 1,000, *recall* sebesar 0,970, dan F1-score sebesar 0,985. Berikut gambar detail confusion matrix dari hasil deteksi:



Gambar 4. 6 Confusion Matrix hasil deteksi

Secara keseluruhan dari gambar 4.6, dari 103 apel yang diproses selama tiga kali pengujian, sistem menghasilkan 99 *true positive* (TP), 3 *false positive* (FP), dan 1 *false negative* (FN). Berdasarkan hasil tersebut, sistem memperoleh nilai *precision* sebesar 0,971, *recall* sebesar 0,990, dan *F1-score* sebesar 0,980. Hasil ini menunjukkan bahwa sistem mampu mempertahankan performa deteksi yang tinggi pada kondisi operasional secara *real-time*, dengan jumlah objek yang terlewat maupun kesalahan deteksi yang relatif rendah.

Analisis lebih lanjut terhadap *confusion matrix* pada Gambar 4.6 menunjukkan bahwa sebagian besar prediksi berada pada diagonal utama, yang menandakan bahwa mayoritas apel berhasil diklasifikasikan ke kelas yang benar. Hanya terdapat beberapa kesalahan klasifikasi antar kelas apel, sedangkan jumlah objek yang tidak terdeteksi (*false negative*) maupun deteksi yang tidak sesuai (*false positive*) relatif sedikit. Kesalahan klasifikasi tersebut kemungkinan besar

dipengaruhi oleh karakteristik dataset yang digunakan selama proses pelatihan model. Dataset terdiri atas apel dengan variasi warna, tingkat kematangan, ukuran, orientasi, kondisi pencahayaan, serta kemiripan visual antar kelas yang cukup tinggi. Selain itu, proses pengambilan dataset buah di dunia nyata memiliki tantangan tersendiri karena sulit memperoleh kondisi objek yang benar-benar seragam. Variasi tersebut menyebabkan batas antar kelas menjadi kurang tegas sehingga beberapa sampel memiliki karakteristik visual yang saling tumpang tindih dan lebih sulit dibedakan oleh model. Oleh karena itu, meskipun model telah mencapai performa yang tinggi, masih terdapat sejumlah kecil kesalahan klasifikasi yang tercermin pada *confusion matrix*.

Nilai *precision*, *recall*, dan *F1-score* pada pengujian *throughput* lebih tinggi dibandingkan hasil evaluasi model TPE pada dataset pengujian. Perbedaan tersebut tidak secara langsung menunjukkan bahwa performa model meningkat setelah diterapkan pada perangkat *embedded*. Perbedaan nilai tersebut terutama dipengaruhi oleh karakteristik objek dan kondisi pengujian yang berbeda. Dataset pengujian model TPE terdiri atas gambar dengan variasi yang lebih beragam, seperti perbedaan varietas apel, warna, tingkat kematangan, kondisi pencahayaan, sudut pengambilan gambar, orientasi objek, ukuran objek, serta latar belakang. Variasi tersebut menyebabkan proses deteksi menjadi lebih kompleks dan meningkatkan kemungkinan terjadinya kesalahan deteksi.

Sebaliknya, apel yang digunakan pada pengujian *real-time* pada alat memiliki kondisi yang lebih terkontrol dibandingkan gambar pada dataset. Proses pengambilan gambar dilakukan menggunakan Raspberry Pi Camera dengan posisi kamera, jarak objek, sudut pengambilan gambar, pencahayaan, arah pergerakan apel, dan latar belakang yang relatif tetap. Kondisi tersebut menghasilkan tampilan objek yang lebih konsisten sehingga karakteristik visual apel yang diterima model lebih seragam. Oleh karena itu, model dapat menghasilkan nilai *confidence* yang lebih stabil dan memperoleh performa deteksi yang lebih tinggi pada pengujian *throughput*.

Dengan demikian, hasil pengujian *throughput* menunjukkan performa sistem secara keseluruhan pada kondisi operasional alat, sedangkan hasil evaluasi model TPE menggunakan dataset menggambarkan kemampuan generalisasi model terhadap variasi gambar yang lebih luas. Kedua hasil pengujian tersebut memiliki tujuan yang berbeda dan tidak dapat dibandingkan secara langsung tanpa mempertimbangkan perbedaan karakteristik data serta kondisi pengujiannya. Pengujian *throughput* digunakan untuk mengevaluasi kemampuan sistem dalam melakukan deteksi dan proses *grading* secara *real-time*, sedangkan pengujian menggunakan dataset digunakan untuk mengevaluasi performa model terhadap data yang memiliki variasi visual yang lebih beragam.