

DETEKSI ANATOMI GIGI MENGGUNAKAN ARSITEKTUR YOLOV8 (YOU ONLY LOOK ONCE) UNTUK PENGENALAN JENIS-JENIS GIGI

Rangga Irsabmalfi dan Reni Rahmadewi

Teknik Elektro, Universitas Singaperbangsa Karawang
Jalan HS. Ronggowaluyo Telukjambe Timur Karawang 41361
2110631160059@student.unsika.ac.id

ABSTRAK

Pemeriksaan gigi secara manual sering kali memakan waktu dan membutuhkan keahlian khusus untuk mengidentifikasi jenis-jenis gigi. Sehingga rentan terjadinya kesalahan manusia karena interpretasi manual oleh dokter atau calon dokter. Oleh karena itu dibutuhkan pendekatan komputerisasi untuk memperkecil terjadi kesalahan dan membantu proses identifikasi untuk diagnosa dini dan kelengkapan perekaman medis. Untuk meningkatkan efisiensi, penelitian ini memanfaatkan algoritma YOLOv8, teknologi *Computer Vision* terkini, untuk mendeteksi dan mengklasifikasikan jenis gigi secara otomatis. Dataset yang digunakan terdiri dari 724 gambar anatomi gigi yang telah dianotasi, mencakup berbagai jenis gigi seperti *molar*, *premolar*, *canine*, dan *incisor*. Model dilatih menggunakan *hyperparameter* seperti 30 dan 50 *epochs* dan ukuran gambar 640x640 piksel. Evaluasi menunjukkan bahwa model memiliki performa tinggi dengan nilai *precision* 94,9%, *recall* 94,6%, *mAP* 97,5%, dan *F1 score* 95% untuk 30 *epochs* dan *precision* 94,9%, *recall* 94,6%, *mAP* 97,5%, dan *F1 score* 95% untuk 50 *epochs*. Hasil ini membuktikan bahwa penggunaan *epochs* tidak mengubah banyak persentase nilai evaluasi jika memiliki komposisi dan jumlah gambar yang sama, selain itu penggunaan YOLOv8 dapat mendeteksi anatomi gigi dengan cepat dan akurat, sehingga dapat menjadi alat bantu yang andal bagi dokter gigi dalam diagnosis dan pemetaan gigi pasien. Penelitian ini juga membuka peluang untuk pengembangan sistem rekam medis digital berbasis teknologi yang lebih efisien.

Kata kunci : Anatomi Gigi, Deteksi Objek, YOLOv8, Pemeriksaan Gigi, Computer Vision

1. PENDAHULUAN

Perkembangan teknologi *Computer Vision* telah membuka peluang baru dalam berbagai bidang, termasuk medis, dengan penerapan algoritma untuk deteksi dan pengenalan objek secara otomatis [1].

Metode berbasis jaringan saraf tiruan (*neural networks*) seperti *Convolutional Neural Networks* (CNN) telah terbukti memberikan hasil yang akurat dalam berbagai tugas deteksi, salah satunya adalah deteksi objek. Deteksi objek digunakan untuk berbagai macam aplikasi, termasuk pada citra medis. Dalam konteks aplikasi medis, penggunaan *Computer Vision* telah mempercepat proses analisis data visual sekaligus mengurangi ketergantungan pada interpretasi manual yang rentan terhadap kesalahan manusia [2].

Salah satu penerapan penggunaan deteksi objek dalam bidang medis adalah deteksi anatomi gigi yang berguna untuk mengetahui jenis-jenis gigi yang ada pada pasien yang ingin melakukan pemeriksaan gigi. Pemeriksaan ini dilakukan untuk pemetaan gigi sebagai acuan tindakan medis selanjutnya. Manfaat lainnya adalah sebagai cara pembelajaran yang mudah untuk mengetahui jenis-jenis gigi pada manusia untuk para calon dokter gigi sehingga lebih mudah mengingat letak dan jenis gigi. Namun, penelitian terkait penggunaan *deep learning* dalam kedokteran gigi masih terbatas, terutama dalam membantu mengenali kondisi gigi tanpa perlu analisis manual yang memerlukan waktu dan kompetensi yang tinggi [3].

Sehingga pendekatan komputerisasi untuk deteksi anatomi gigi pasien dapat membantu dokter dan calon dokter dalam proses identifikasi untuk kebutuhan diagnosa dini dan kelengkapan perekaman medis [3] [4].

Deteksi objek memiliki beberapa tantangan, salah satunya adalah keragaman bentuk, ukuran, dan orientasi objek yang dapat berbeda pada setiap gambar. Selain itu, pencahayaan, posisi, serta kualitas gambar dapat memengaruhi keakuratan hasil deteksi [5].

Oleh karena itu, dibutuhkan sebuah model yang mampu mengatasi berbagai variabilitas tersebut dengan performa yang handal. Salah satu pendekatan yang dianggap efektif dalam mengatasi masalah ini adalah penggunaan arsitektur YOLO (*You Only Look Once*), terutama versi terbaru, yaitu YOLOv8, yang dikenal memiliki akurasi dan kecepatan tinggi dalam mendeteksi objek.

Arsitektur YOLOv8 dipilih dalam penelitian ini karena kemampuannya dalam melakukan deteksi objek secara *real-time*, dengan akurasi yang lebih baik dan proses yang lebih cepat dibandingkan pendahulunya [6].

YOLOv8 dilengkapi dengan berbagai penyempurnaan pada bagian *backbone* dan *head detection* yang memungkinkan sistem mendeteksi objek dengan lebih teliti, bahkan dalam kondisi pencahayaan dan sudut pandang yang bervariasi. Keunggulan ini sangat bermanfaat untuk mendeteksi jenis gigi pada gambar gigi pasien yang umumnya memiliki resolusi tinggi namun memerlukan analisis

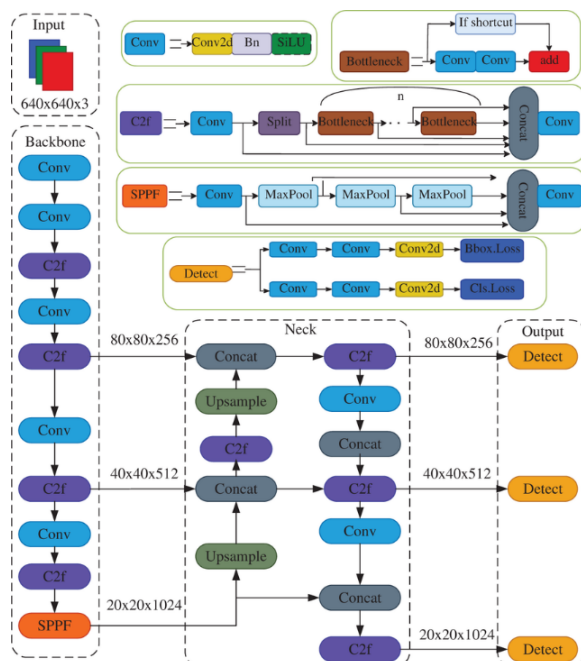
cepat. Melalui YOLOv8, setiap jenis gigi yang terdapat dalam peta gigi dapat diidentifikasi dan diklasifikasikan secara otomatis.

Penggunaan deteksi jenis gigi dengan YOLOv8 tidak hanya mempercepat proses diagnosis, tetapi juga memberikan potensi efisiensi dalam pemetaan gigi pasien. Deteksi jenis gigi yang akurat sangat penting bagi dokter gigi dalam menyusun rencana perawatan yang tepat sesuai dengan jenis gigi dan kondisinya. Selain itu, teknologi ini memungkinkan perekaman data secara digital yang terstruktur sehingga data riwayat kondisi gigi pasien dapat diakses dengan lebih mudah dan dapat mendukung pengembangan sistem rekam medis digital dalam jangka panjang.

Secara keseluruhan, penelitian ini bertujuan untuk mengembangkan model deteksi anatomi gigi berbasis YOLOv8 yang mampu mendeteksi dan mengklasifikasikan jenis gigi pasien secara efektif dan efisien. Dengan memanfaatkan YOLOv8 sebagai arsitektur deteksi, diharapkan sistem yang dihasilkan dapat berfungsi sebagai alat bantu yang andal bagi dokter gigi dalam mengidentifikasi dan memetakan jenis gigi dengan cepat dan akurat.

2. TINJAUAN PUSTAKA

2.1. Arsitektur YOLOv8



Gambar 1. Arsitektur YOLOv8

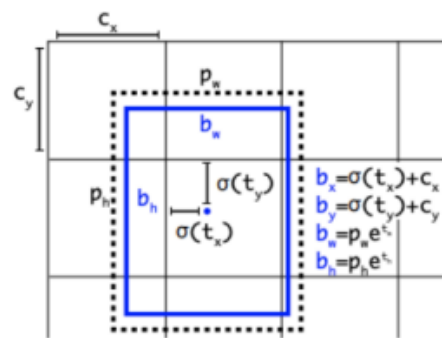
(Sumber: Hiangfu, et al. 2024)

Dalam penelitian ini menggunakan arsitektur YOLOv8 (*You Only Look Once*) yang dirilis pada tahun 2023 [7] sebagai metode penelitiannya. Algoritma YOLOv8 adalah pengembangan terbaru dari algoritma YOLO sebelumnya yang berbasis pada *Convolutional Neural Network* (CNN) yang dibuat oleh *ultralytics* [8]. YOLO, singkatan dari *You Only Look Once*, merupakan salah satu algoritma deteksi

objek yang mampu mengidentifikasi objek dalam gambar atau video secara *real-time*.

Arsitektur YOLOv8, seperti yang ditampilkan pada Gambar 1, terdiri atas tiga komponen utama: jaringan *backbone*, *neck*, dan *head*. Jaringan *backbone* menggunakan *Feature Pyramid Network* (FPN) untuk mengekstraksi fitur dari gambar masukan. Sementara itu, *neck* memanfaatkan serangkaian *cross-layer connection* untuk meningkatkan kualitas fitur yang diekstraksi. Komponen *head* kemudian menggunakan fitur yang telah ditingkatkan ini untuk memprediksi kotak pembatas, skor kelas objek, serta tingkat akurasi setiap objek dalam gambar [9].

Model ini menggunakan *anchor-free detection*, yang memungkinkan prediksi langsung pada pusat objek tanpa bergantung pada kotak jangkar (*anchor box*), seperti yang ditunjukkan pada Gambar 2 [10].



Gambar 2. Anchor Box

(Sumber: Y. T. Wulandari dan T. Rosandy. 2024)

Secara umum, YOLO membagi gambar *input* menjadi beberapa *grid* yang terdiri dari sel-sel. Setiap sel dalam *grid* ini bertugas memprediksi beberapa *bounding box*, beserta kemungkinan kelas objek yang mungkin ada di dalamnya [8].

2.2. Google Colab



Gambar 3. Logo Google Colab

(Sumber: colab.google.com)

Pada Gambar 3 merupakan logo dari Google Colab yang akan digunakan dalam penelitian ini. Google Colab adalah platform *cloud computing* dari Google yang memungkinkan pengguna menjalankan kode Python langsung di *browser* tanpa perlu instalasi lokal. Platform ini banyak digunakan untuk pemrosesan data, pelatihan model kecerdasan buatan, dan analisis data [11].

2.3. Python



Gambar 4. Logo Python
(Sumber: python.org)

Pada Gambar 4 merupakan logo dari Bahasa pemrograman Python yang akan digunakan dalam penelitian ini. Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dan pertama kali dirilis pada tahun 1991. Saat ini, Python menjadi salah satu bahasa pemrograman yang sangat populer. Python juga bersifat serbaguna, dengan berbagai aplikasi seperti *machine learning* dan *deep learning*. Python memiliki penulisan sintaks yang mudah, *library* yang lengkap, dan komunitas yang kuat karena bersifat *open source*. Ada banyak IDE yang dapat menuliskan bahasa Python, mulai dari VSCode, Pycharm, Jupyter Notebook, dan Google Colab[12].

2.4. Precision

Precision adalah rasio antara *True Positive* (TP) dengan total data yang diprediksi sebagai positif, yaitu gabungan antara *True Positive* (TP) dan *False Positive* (FP), seperti yang ditunjukkan pada Persamaan 1. *Precision* menggambarkan tingkat keakuratan hasil prediksi model dalam mengidentifikasi data yang sesuai dengan yang diharapkan[13].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

2.5. Recall

Recall adalah rasio antara *True Positive* (TP) dengan total data sebenarnya yang positif, yaitu gabungan antara *True Positive* (TP) dan *False Negative* (FN), seperti yang ditunjukkan pada Persamaan 2. *Recall* menggambarkan sejauh mana model berhasil menemukan kembali informasi yang relevan[13].

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

2.6. Mean-Average Precision (mAP)

Mean Average Precision (mAP) adalah nilai rata-rata dari *Average Precision* (AP) untuk setiap kelas objek. Perhitungan mAP dilakukan menggunakan persamaan seperti yang ditunjukkan pada Persamaan 3. Dalam persamaan tersebut, N menunjukkan jumlah total kelas yang dilatih, i adalah indeks iterasi yang dimulai dari 1, dan AP merupakan *Average Precision*[13].

$$mAP = \frac{1}{N} \sum_{k=1}^{k=N} AP_k \times 100\% \quad (3)$$

2.7. F1 Score

F1 Score adalah rata-rata harmonis antara *precision* dan *recall*, seperti yang dijelaskan pada Persamaan 4. *F1 Score* mencapai nilai optimal ketika model memiliki *precision* dan *recall* yang tinggi. Nilai terbaik *F1 Score* adalah 1.0, sedangkan nilai terendah adalah 0.0. *F1 Score* sering digunakan pada dataset yang menghadapi masalah ketidakseimbangan, yaitu ketika jumlah *False Negative* (FN) dan *False Positive* (FP) tidak proporsional [13].

2.8. Confusion Matrix

Confusion matrix adalah tabel yang memberikan gambaran perbandingan antara hasil klasifikasi model dengan klasifikasi sebenarnya. *Confusion matrix* digunakan untuk mengevaluasi kinerja model, seperti yang ditunjukkan pada Gambar 7.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 5. Tabel *Confusion Matrix*
(Sumber: H. Nurhadiati, et al. 2022)

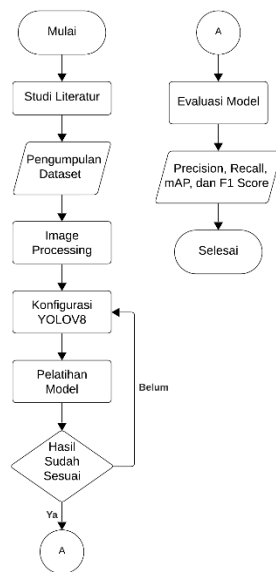
Pada Gambar 5, dapat dilihat bahwa matriks ini membandingkan hasil prediksi model dengan nilai aktual dan terdiri dari empat komponen utama, yaitu:

- True Positive (TP) yang menunjukkan data aktual positif diprediksi positif oleh model (prediksi benar).
- False Positive (FP) yang menunjukkan data aktual negatif diprediksi positif oleh model (prediksi salah, dikenal juga sebagai *Type I Error*).
- False Negative (FN) yang menunjukkan data aktual positif diprediksi negatif oleh model (prediksi salah, dikenal juga sebagai *Type II Error*).
- True Negative (TN) yang menunjukkan data aktual negatif diprediksi negatif oleh model (prediksi benar)[13].

3. METODE PENELITIAN

3.1. Alur Penelitian

Secara umum, alur penelitian yang akan dibuat dijelaskan dengan *flowchart* pada Gambar 5. Metode penelitian pada Gambar 6 menjabarkan tahapan-tahapan yang akan dilakukan dalam proses pendeteksian anatomi gigi.



Gambar 6. Alur Penelitian

3.2. Dataset

Dataset adalah sekumpulan objek yang dilengkapi dengan atributnya. Atribut mengacu pada karakteristik atau sifat yang dimiliki oleh suatu objek [14]. *Dataset* yang digunakan dalam penelitian ini adalah *dataset* citra anatomi gigi yang sudah dianotasi untuk penggunaan YOLOV8 yang didapatkan melalui *website kaggle* seperti pada Gambar 7 yang berisi citra gigi dari berbagai orang dalam berbagai sudut pandang seperti tampak depan dan bagian oral yang menunjukkan beberapa citra gigi yang diberi label berdasarkan jenisnya seperti *1st molar* (gigi geraham pertama), *2nd molar* (gigi geraham kedua), *1st premolar* (gigi premolar pertama), *2nd premolar* (gigi premolar kedua), *canine* (gigi taring), *central incisor* (gigi seri tengah), *lateral incisor* (gigi seri pinggir) [15].

Gambar 7. Dataset Anatomi Gigi
(Sumber: kaggle.com)

Dari dataset tersebut terdapat 724 gambar yang telah dilakukan *image pre-processing* dan dilabeli dengan label sebelumnya yang terdiri dari data *train*, *test*, dan *validation* yang masing-masing memiliki jumlah gambar sebagai berikut [13].

Tabel 1. Jumlah Gambar Dataset Anatomi Gigi

Jenis	Jumlah
<i>Train</i>	505 gambar
<i>Test</i>	112 gambar
<i>Validation</i>	107 gambar

Berdasarkan pada Tabel 1, jumlah dari gambar *train* terdiri dari 505 gambar atau 70% dari total yang akan dipakai saat *training* model dengan YOLOV8 yang nantinya akan hasil dari model tersebut akan diuji dengan 112 gambar atau 15% dari total yang ada di *test*, lalu akan divalidasi hasil *test* tersebut dengan 15% sisanya atau 107 gambar yang ada di *validation*.

3.3. Image Pre-Processing

Proses *Image Pre-Processing* akan dilakukan anotasi citra dengan memberikan *bounding box* dan *label* pada masing-masing gambar gigi yang telah diklasifikasikan berdasarkan lokasi dan jenis gigi yang dimiliki dengan menggunakan *Roboflow* [16].

Selain itu gambar yang telah dianotasi akan diubah ukurannya menjadi 640x640 *pixel* atau diambil bagian giginya saja agar sesuai dengan ketentuan penggunaan model YOLOV8 agar memberikan hasil yang optimal dan akurat saat *training*.

3.4. Konfigurasi YOLOv8

Setelah proses *image pre-processing* selesai, langkah berikutnya adalah mempersiapkan dan mengkonfigurasi model YOLOv8. Tahap ini dilakukan di Google Colab, mencakup pengecekan sumber daya perangkat keras akselerator, instalasi pustaka yang dibutuhkan, impor *dataset*, konfigurasi *hyperparameter*, dan pelatihan model.

Pengecekan Google Colab bertujuan untuk memastikan ketersediaan perangkat keras akselerator seperti GPU. *Library* Ultralytics diinstal untuk mengakses model YOLOv8, sedangkan Matplotlib digunakan untuk membuat grafik dan menampilkan hasil deteksi objek. Selain itu, NumPy digunakan untuk memanipulasi *array* data citra, seperti mengubah format *array* gambar atau melakukan transformasi matematis pada data.

3.5. Pelatihan Model

Model deteksi anatomi gigi dilatih dengan menggunakan bahasa pemrograman Python pada Google Colab untuk mengeksekusi *code* yang nilai *hyperparameter epoch* sebesar 50 kali, *image size* 640x640 untuk evaluasi, dan ambang batas *confidence* 0.25.

Hasil dari model yang dilatih akan disimpan dalam *directory* Google Drive yang terhubung dengan

Google Colab sehingga dapat diakses kembali tanpa perlu pelatihan ulang dan dapat diuji kembali.

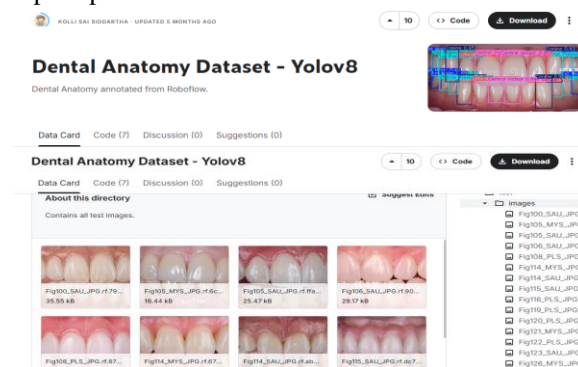
3.6. Evaluasi dan Pengujian Model

Pengujian model dilakukan langsung dalam Google Colab dengan mengakses model yang telah dilatih yang berada di *folder* Google Drive. Hasil pengujian akan dievaluasi dengan metrik evaluasi seperti *Precision*, *Recall*, *Mean Average Precision* (mAP), *F1 Score*.

4. HASIL DAN PEMBAHASAN

4.1. Persiapan Dataset

Dataset yang digunakan untuk membuat model deteksi anatomi gigi ini adalah *dataset* yang sebelumnya telah diambil melalui *website* kaggle.com seperti pada Gambar 8 berikut :

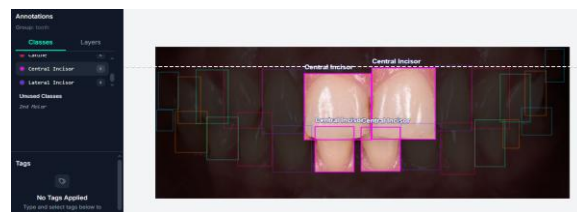


Gambar 8. Persiapan Dataset

Pada Gambar 8 dapat terlihat bahwa *dataset* yang memiliki banyak gambar gigi yang siap digunakan sehingga dapat dilakukan proses *image pre-processing* agar ketika dilatih dapat menghasilkan hasil yang lebih baik dan semakin akurat.

4.2. Image Pre-Processing

Setelah *dataset* didapatkan, selanjutnya adalah melakukan *image pre-processing* dengan menggunakan Roboflow untuk melakukan anotasi, klasifikasi, penyesuaian ukuran gambar agar semuanya berukuran 640x640 *pixel* atau diambil bagian giginya saja seperti pada Gambar 9 berikut.

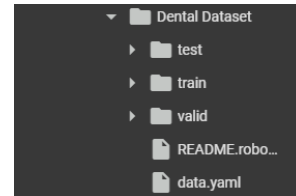


Gambar 9. Image Pre-Processing dengan Roboflow

Pada gambar di atas menunjukkan bahwa gambar tersebut diambil hanya bagian giginya saja lalu anotasi dilakukan dengan pemberian kotak batas (*bounding box*) dan diberi label sesuai dengan nama gigi yang sesuai. Ketika sudah diberikan label dan *bounding box* yang sesuai, maka selanjutnya adalah memilah gambar-gambar tersebut ke dalam *folder* *train*, *test*,

dan *valid* serta mengubah formatnya untuk penggunaan YOLOV8 agar dapat digunakan untuk melatih.

Selanjutnya, *dataset* tersebut di-download lalu di-upload ke Google Colab untuk digunakan sebagai *input* dalam melatih model tersebut seperti yang ditunjukkan pada Gambar 10 berikut.

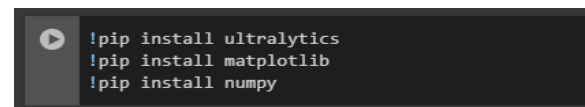


Gambar 10. Persiapan Dataset ke Google Colab

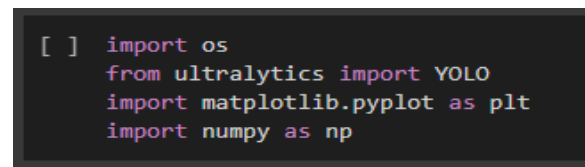
Pada Gambar 10 terlihat bahwa *dataset* “Dental Dataset” berisi tiga *folder* yang berguna dalam proses pelatihan model nanti serta *data.yaml* untuk keperluan yang sama. Setelah di-upload, maka dapat dilanjutkan untuk melatih modelnya.

4.3. Konfigurasi YOLOV8

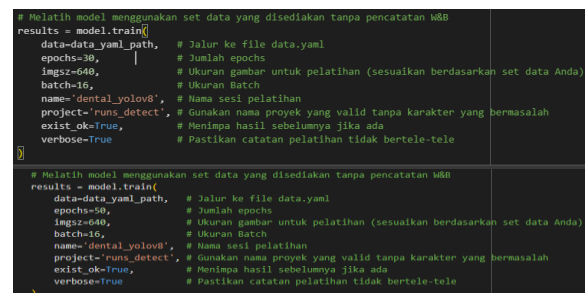
Dari hasil pengujian yang telah dilakukan yaitu dengan menggunakan model YOLOv8 sebagai untuk melatih model deteksi anatomi gigi dengan Google Colab sebagai IDE yang digunakan. Tahap awal yang dilakukan dalam melatih model adalah menentukan perangkat keras akselerator yang dipakai, dalam hal ini peneliti menggunakan GPU bawaan dari Google Colab untuk melatih modelnya. Selanjutnya menentukan seberapa banyak *epochs* yang dipakai, peneliti menggunakan 30 dan 50 *epochs* seperti pada Gambar 13 untuk membandingkan besar nilai *mAP*, *recall*, dan *precision* dari model deteksi ini.



Gambar 11. Menginstalasi Library



Gambar 12. Memanggil Library



Gambar 13. Konfigurasi Hyperparameter (30 dan 50 Epochs)

Berdasarkan Gambar 11 dan 12, dilakukan instalasi tiga *library*, yaitu *ultralytics*, *matplotlib*, dan *numpy* dan memanggil *YOLO*, *matplotlib.pyplot* dan *numpy* untuk membuat model dan memvisualisasikan hasilnya dalam bentuk plot agar lebih mudah dievaluasi. Selain itu dalam pelatihan model ini digunakan parameter seperti pada Tabel 2 berikut.

Tabel 2. Parameter Pelatihan Model

Percobaan	Parameter	Nilai
1	<i>Epochs</i>	30
	<i>Image Size</i>	640
	<i>Batch</i>	10
2	<i>Epochs</i>	50
	<i>Image Size</i>	640
	<i>Batch</i>	10

4.4. Pelatihan Model Anatomi Gigi

Saat sudah ditentukan parameter pada masing-masing percobaan untuk melatih model deteksi anatomi gigi, maka tinggal *run* sintaks yang telah ditulis untuk melakukan pelatihan model seperti pada Gambar 14 dan 15 yang menunjukkan proses *training* menggunakan 505 gambar *train* yang dilakukan.

Gambar 14. Pelatihan Model 30 Epochs

Gambar 15. Pelatihan Model 50 Epochs

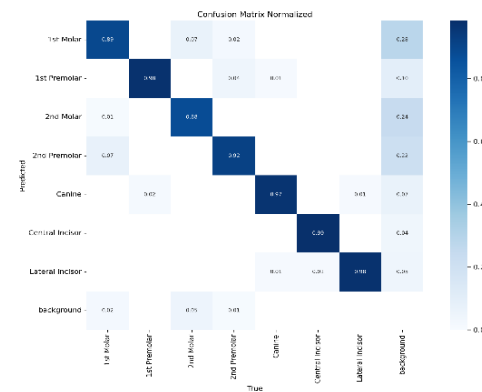
Tabel 3. Evaluasi Model Deteksi Anatomi Gigi 30 Epochs

No	Class	Precision	mAP	Recall
1	All	0.941	0.970	0.939
2	1 st Molar	0.918	0.959	0.885
3	1 st Premolar	0.955	0.979	0.979
4	2 nd Molar	0.835	0.905	0.809
5	2 nd Premolar	0.93	0.973	0.945
6	Canine	0.974	0.987	0.975
7	Central Incisor	0.995	0.995	0.995
8	Lateral Incisor	0.978	0.994	0.983

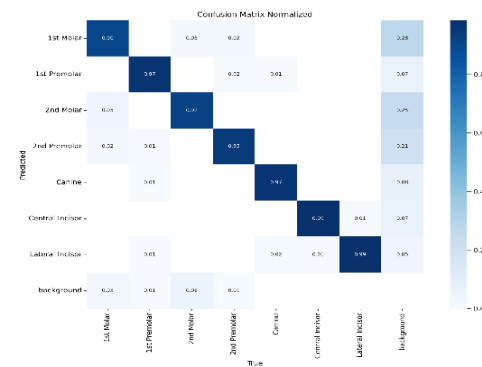
Tabel 4. Evaluasi Model Deteksi Anatomi Gigi 50 Epochs

No	Class	Precision	mAP	Recall
1	All	0.949	0.975	0.946
2	1 st Molar	0.938	0.957	0.894
3	1 st Premolar	0.966	0.986	0.968
4	2 nd Molar	0.859	0.929	0.873
5	2 nd Premolar	0.94	0.976	0.934
6	Canine	0.984	0.991	0.973
7	Central Incisor	0.987	0.993	0.993
8	Lateral Incisor	0.971	0.988	0.988

Hasil dari pelatihan model tadi juga menghasilkan grafik visualisasi *confusion matrix* pada Gambar 16 dan 17 sebagai berikut.



Gambar 16. Hasil Confusion Matrix 30 Epochs



Gambar 17. Hasil Confusion Matrix 50 Epochs

4.5. Evaluasi Model Anatomi Gigi

Setelah dilatih dengan model YOLOv8, maka akan diuji oleh gambar gigi yang ada pada *folder test* sebanyak 112 gambar dan didapatkan hasil *precision*, *mAP*, dan *recall* untuk masing-masing *class* pada model deteksi anatomi gigi.

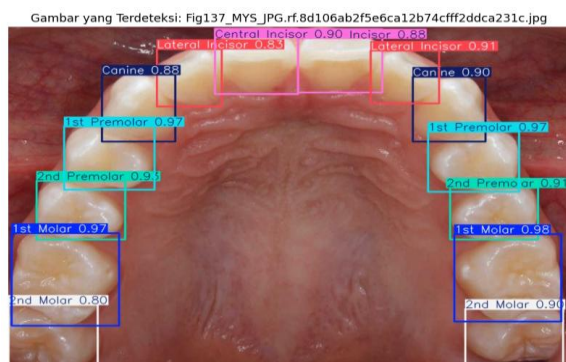
Tabel 3 dan 4 merupakan hasil validasi model dari kedua percobaan yang menunjukkan secara keseluruhan model telah dilatih dengan baik dan mampu mendeteksi anatomi gigi dengan nilai *precision* yang cukup tinggi.

Untuk masing-masing dari kelas dari klasifikasi jenis gigi yang telah dilatih dengan menggunakan kedua *epoch* yang berbeda memberikan hasil yang mirip meskipun berbeda, seperti untuk *lateral* dan *central incisor* memiliki nilai *precision*, *recall*, dan *mAP* yang lebih tinggi dibandingkan dengan 2nd *molar* yang memiliki ketiga nilai yang rendah. Hal ini disebabkan karena gigi incisor atau gigi seri lebih mudah terlihat karena berada di paling depan sedangkan gigi molar atau geraham berada di belakang sehingga lebih sulit untuk terlihat ketika deteksi berlangsung.

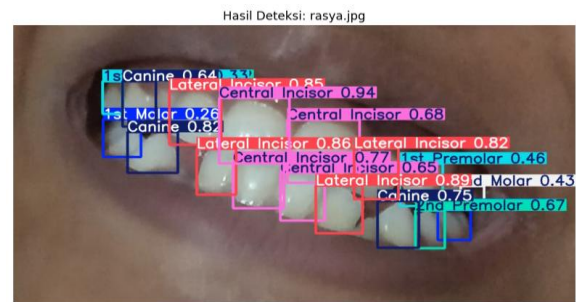
Secara keseluruhan hasil pelatihan model dengan menggunakan 30 *epochs* dalam mendeteksi seluruh anatomi gigi memiliki total nilai *precision* sebesar 94,1%, nilai *recall* sebesar 93,9%, nilai *mAP* sebesar 97%, dan nilai *F1 score* sebesar 94%, sedangkan jika menggunakan 50 akan menghasilkan nilai *precision* sebesar 94,9%, nilai *recall* sebesar 94,6%, nilai *mAP* sebesar 97,5%, dan nilai *F1 score* sebesar 95%. Hal ini menunjukkan bahwa model deteksi anatomi gigi yang menggunakan 50 *epochs* memiliki kemampuan untuk mendeteksi jenis-jenis gigi yang lebih akurat daripada yang menggunakan 30 *epochs*, kemudian data tes yang telah dibagi sebelumnya digunakan untuk melakukan prediksi anatomi gigi untuk menguji hasil dari data *training* dan data validasi yang digunakan sebelumnya. Berikut merupakan hasil prediksi dari data tes dan data dari pengujian langsung yang dilakukan.



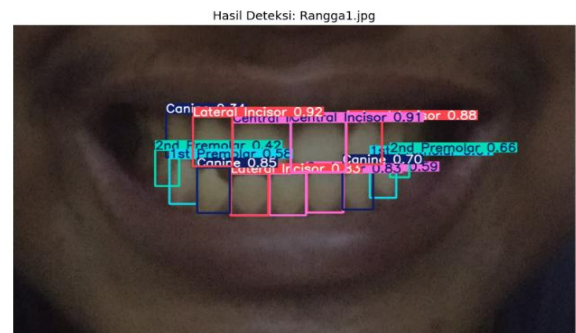
Gambar 18. Hasil Prediksi Data Test 30 Epochs



Gambar 19. Hasil Prediksi Data Test 50 Epochs



Gambar 20. Hasil Prediksi Data Langsung 30 Epochs



Gambar 21. Hasil Prediksi Data Langsung 50 Epochs

Model YOLOv8 berfungsi dengan cara membagi gambar menjadi beberapa kotak pembatas (*bounding box*) untuk mengidentifikasi objek, dalam hal ini gigi-gigi di dalam mulut. Gambar 18 sampai 21 menunjukkan contoh hasil prediksi pada data *test*, yang didapatkan setelah proses pelatihan dan validasi model. Hasil klasifikasi gigi dengan nilai *confidence* yang tinggi menunjukkan bahwa model memiliki tingkat kepastian yang baik dalam mendeteksi anatomi gigi.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan dengan menggunakan 724 gambar dataset yang dengan komposisi 70% data *train*, 15% data *test*, dan 15% data valid, maka model deteksi anatomi gigi dapat dilatih dengan menggunakan *hyperparameter epochs* 30 dan 50 dan memiliki *batch* 10, dan *image size* 640 sama. Hasil dari pelatihan kedua model tersebut memiliki nilai evaluasi keseluruhan seperti *presicion* sebesar 94,1%, nilai *recall* sebesar 93,9%, nilai *mAP* sebesar 97%, dan nilai *F1 score* sebesar 94% untuk 30 *epochs* dan *presicion* sebesar 94,9%, nilai *recall* sebesar 94,6%, nilai *mAP* sebesar 97,5%, dan nilai *F1 score* sebesar 95% untuk 50 *epochs*. Hasil ini menunjukkan bahwa perbedaan *epochs* pada saat pelatihan model hanya sedikit jika menggunakan banyak gambar dan komposisi yang sama. Selain itu, YOLOv8 dapat menjadi alat bantu yang efektif bagi dokter gigi dalam mengidentifikasi dan memetakan jenis gigi pasien dengan cepat dan akurat. Selain itu, teknologi ini juga berpotensi untuk meningkatkan sistem rekam medis digital dengan menyediakan data yang terstruktur dan mudah diakses.

Untuk penelitian selanjutnya pada deteksi anatomi gigi ini dapat focus pada penggunaan secara *real-time* dan menggunakan arsitektur deteksi objek yang lebih optimal seperti YOLOv11 atau Faster R-CNN yang dapat memberikan hasil deteksi yang lebih baik.

DAFTAR PUSTAKA

- [1] S. Susandri, S. Sumijan, A. Zamsuri, R. Rahmiati, and A. Asparizal, "Enhancing Dental Image Segmentation Techniques: Edge Detection and Color Thresholding", *Digitalzone*, vol. 15, no. 1, pp. 94-104, May 2024.
- [2] N. H. Danial and D. Setiawati, "Convolutional Neural Network (CNN) Based On Artificial Intelligence In Periodontal Diseases Diagnosis", *interdental*, vol. 20, no. 1, pp. 139-148, Apr. 2024.
- [3] Supiyandi, W. E. Judistira, S. Nurliani, R. S. Darmono, and Inneke Putri, "Penerapan Deep Learning Dalam Analisis Citra Gigi", *JUPENDIS (Jurnal Pendidikan dan Ilmu Sosial)*, vol. 2, no. 4, pp. 117-128, Oct. 2024, doi: <https://doi.org/10.54066/jupendis.v2i4.2165>
- [4] Rika Amran, Anisah Apriyani, and N. P. Dewi, "Peran Penting Kelengkapan Rekam Medik di Rumah Sakit", *Baiturrahmah Medical Journal*, vol. 1, no. 1, pp. 69-76, 2021, Accessed: Mar. 22, 2025. <https://jurnal.unbrah.ac.id/index.php/brmj/article/view/1061>
- [5] Dzaky Nafis Alfarizi, R. Agung Pangestu, Dimas Aditya, Muhammad Adi Setiawan, and P. Rosyani, "Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis", *aidanspk*, vol. 1, no. 1, pp. 54-63, Jun. 2023.
- [6] Zhongmin Huangfu, S. Li, and L. Yan, "Ghost-YOLO v8: An Attention-Guided Enhanced Small Target Detection Algorithm for Floating Litter on Water Surfaces", *Computers, materials & continua/Computers, materials & continua (Print)*, vol. 80, no. 3, pp. 3713-3731, Jan. 2024, doi: <https://doi.org/10.32604/cmc.2024.054188>.
- [7] M. Sohan, Thotakura Sai Ram, and V. Rami, "A Review on YOLOv8 and Its Advancements", *ResearchGate*, Jan. 07, 2024. https://www.researchgate.net/publication/377216968_A_Review_on_YOLOv8_and_Its_Advan_cements
- [8] Imam Maulana, Nining Rahaningsih, and Tati Suprapti, "Analisis Penggunaan Model YOLOv8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3621-3627, Feb. 2024, doi: <https://doi.org/10.36040/jati.v7i6.8271>.
- [9] Yanto, F. Aziz, Irmawati, "YOLO-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1437-1444, Jun. 2023.
- [10] Y. T. Wulandari, T. Rosandy, "Implementasi Computer Vision Dalam Sistem Deteksi Gerakan Disiplin Kampus," *Jurnal TEKNIKA*, vol. 18(2), pp. 343-354, 2024
- [11] R. Andarsyah and Amri Yanuar, "Sentimen Analisis Aplikasi Posaja Pada Google Playstore Untuk Peningkatan Pospay Superapp Menggunakan Support Vector Meachine," *Jurnal Teknik Informatika*, vol. 16, no. 2, pp. 1-7, 2024, Accessed: Nov. 20, 2024. [Online]. Available: <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/3533#>
- [12] M. R. S. Alfarizi, M. Z. . Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. . Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning", *karimahtauhid*, vol. 2, no. 1, pp. 1-6, Jan. 2023.
- [13] H. Nurhadiati, S. A. Wibowo, A. Pratondo, "Analisis Performansi Deteksi Objek Pada Metode Complex YOLOv4 Untuk Autonomous Driving," *e-Prossiding of Engineering*, pp. 2753-2758, 2022.
- [14] E. Listiana and M. A. Muslim, "Penerapan Adaboost Untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi Pada Diagnosa Chronic Kidney Disease.," *Prosiding SNATIF*, pp. 875-881, 2017.
- [15] Satria Aji Purwoko, "Mengenal Anatomi Gigi, Jenis, dan Fungsi Bagiannya," *hallosehat.com*, 2024. <https://hallosehat.com/gigi-mulut/anatomi-gigi-dan-fungsinya/>
- [16] Kolli Sai Siddhartha, "Dental Anatomy Dataset - Yolov8," *Kaggle.com*, 2024. <https://www.kaggle.com/datasets/saisiddhartha69/dental-anatomy-dataset-yolov8>
- [17] Roboflow, "Preprocess Images | Roboflow Docs," *Roboflow.com*, May 23, 2024. <https://docs.roboflow.com/datasets/image-preprocessing> (accessed Nov. 20, 2024).