

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Makanan Olahan

Makanan olahan adalah makanan yang telah mengalami proses pengolahan dari bentuk aslinya dengan tujuan meningkatkan cita rasa, keamanan, daya simpan, atau kemudahan dalam penyajian. Proses tersebut dapat memengaruhi kandungan gizi dan jumlah kalori dibandingkan dengan bahan pangan aslinya. Berdasarkan sistem klasifikasi Nova yang dikemukakan oleh Monteiro dkk., 2016, makanan dibagi menjadi empat kelompok berdasarkan tingkat pengolahannya, yaitu :

1. *Minimally Processed Food* (Makanan minim proses)

Kelompok ini terdiri dari bahan pangan alami yang hanya mengalami proses pengolahan sederhana seperti penyucian, pengupasan, pemotongan, perebusan, pembekuan dan lainnya. Contohnya seperti buah-buahan, nasi, sayuran, atau telur rebus.

2. *Processed Culinary Ingredients* (Bahan tambahan kuliner)

Merupakan bahan yang digunakan untuk proses memasak untuk membumbui atau mengolah makanan. Contohnya seperti garam, gula, dan minyak nabati.

3. *Processed Food* (Makanan olahan).

Makanan ini dibuat dengan menambahkan garam, gula, atau zat lain ke makanan alami. Tujuannya adalah untuk meningkatkan daya simpan atau memperbaiki cita rasa makanan. Contohnya seperti ikan goreng, dendeng, ayam goreng, tahu goreng, ayam bakar, tempe goreng dan masih banyak lagi.

4. *Ultra-processed Food* (Makanan ultra-olahan).

Kelompok ini merupakan produk industri seperti mi instan dan *nugget* yang diformulasikan menggunakan sedikit atau tanpa bahan makanan utuh, serta mengandung aditif kosmetik dan rekayasa sensorik.

2.1.2 Kalori

Kalori adalah nilai energi yang terkandung di dalam makanan dan minuman yang kita konsumsi (Astuti dkk., 2021). Kegunaan dari kalori adalah sebagai

sumber energi kita dalam melakukan berbagai aktivitas seperti bergerak, mencerna makanan, menjaga suhu tubuh, dan lain sebagainya. Tidak hanya itu, kalori juga dapat menjadi salah satu faktor kesehatan dan pembentukan tubuh.

Kebutuhan kalori tiap orang dapat berbeda-beda dikarenakan beberapa faktor seperti usia, jenis kelamin, tingkat aktivitas, dan kondisi kesehatan. Jika jumlah kalori yang dikonsumsi sesuai dengan kebutuhan, maka tubuh akan berada dalam kondisi seimbang. Namun, jika terjadi ketidakseimbangan (baik kelebihan ataupun kekurangan kalori) maka dapat berdampak pada berat badan dan kesehatan secara keseluruhan.

2.1.3 Deep Learning

Deep Learning merupakan cabang dari pembelajaran mesin (*Machine Learning*) yang memanfaatkan jaringan saraf tiruan (*artificial neural networks*) (Ilahiyah & Nilogiri, 2018). Dengan jaringan saraf tiruan ini, memungkinkan *deep learning* untuk mempelajari pola dan representasi data yang kompleks sehingga biasanya digunakan untuk mempelajari pola pada gambar. Model *deep learning* bekerja dengan memanfaatkan lapisan-lapisan dalam jaringan saraf yang dapat mengenali pola secara hierarkis, mulai dari fitur sederhana hingga yang lebih kompleks.

Saat ini, model *deep learning* dapat dilatih dengan *dataset* yang besar, sehingga memungkinkan pengembangan sistem yang lebih akurat dan efisien. Ini dikarenakan kemajuan teknologi pada bagian *hardware* seperti GPU (*Graphics Processing Unit*) yang mampu mempercepat proses komputasi dengan memproses banyak operasi secara paralel dan TPU (*Tensor Processing Unit*) yang dirancang khusus untuk mempercepat operasi terkait tensor dalam model *deep learning*. Dengan kemajuan ini, teknologi *deep learning* dapat diterapkan dalam berbagai bidang seperti pengenalan wajah, pemrosesan bahasa alami, serta diagnosis medis.

2.1.4 Machine Learning

Machine learning adalah cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pembelajaran data. Menurut Golberg dan Holland, *machine learning* merupakan penerapan algoritma matematika dan program komputer yang dirancang untuk belajar dari data yang tersedia, sehingga mampu

membuat prediksi untuk kejadian di masa depan (Roihan dkk., 2019). Dari definisi tersebut, proses pembelajaran dalam *machine learning* melibatkan tiga komponen utama, yaitu *dataset* (kumpulan data yang digunakan untuk pelatihan), algoritma pembelajaran (metode yang digunakan untuk menemukan pola dari data), dan model (hasil pembelajaran yang digunakan untuk prediksi).

Machine learning terbagi menjadi dua kategori utama, yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* adalah pendekatan di mana model dilatih menggunakan *dataset* yang sudah dilabeli atau dapat dikatakan di setiap data *input* memiliki *output* yang diketahui, biasanya digunakan dalam melakukan prediksi pada data baru. Sementara itu, *unsupervised learning* digunakan ketika data yang tersedia tidak memiliki label sehingga model harus menemukan pola, struktur, atau kelompok dalam data secara mandiri oleh karena itu biasanya kategori ini digunakan untuk melakukan pengelompokan.

2.1.5 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang dirancang untuk memproses data seperti gambar dan video. Arsitektur CNN akan mengidentifikasi pola kompleks yang ada pada gambar untuk mendeteksi pola spasial pada citra seperti garis tepi, tekstur, hingga bentuk objek (Hadi dkk., 2024). CNN bekerja dengan menggunakan beberapa lapisan utama yang bekerja secara berurutan untuk mengekstrak fitur dari gambar *input*. Lapisan-lapisan utama dalam CNN ini meliputi:

1. Convolutional Layer

Lapisan ini memiliki fungsi untuk mengekstraksi fitur dari gambar dengan menggunakan filter (kernel). Setiap filter akan mendeteksi pola tertentu yang ada didalam gambar, seperti garis tepi atau tekstur.

2. Activation Function

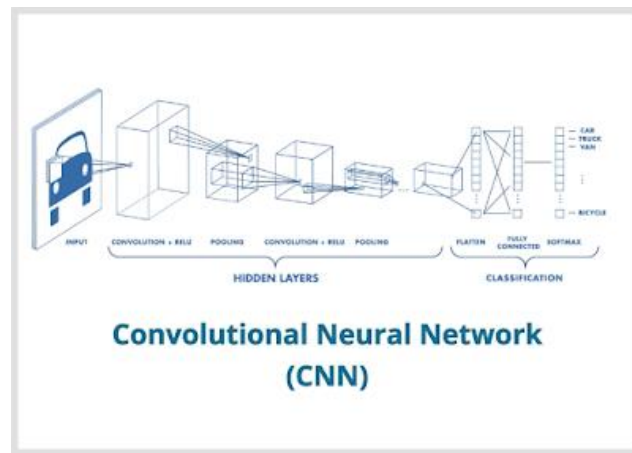
Fungsi aktivitas digunakan untuk membuat algoritma CNN dapat menangkap pola yang lebih kompleks. Contohnya adalah ReLU (*Rectified Linear Unit*) yang berfungsi untuk meningkatkan non-linearitas dalam model.

3. Pooling layer

Lapisan ini berfungsi untuk mengurangi dimensi fitur yang dihasilkan dari lapisan konvolusi, sehingga mengurangi jumlah parameter dan meningkatkan efisiensi komputasi. Contoh jenis pooling yang sering digunakan adalah *max pooling* dan *average pooling*.

4. *Fully Connected Layer*

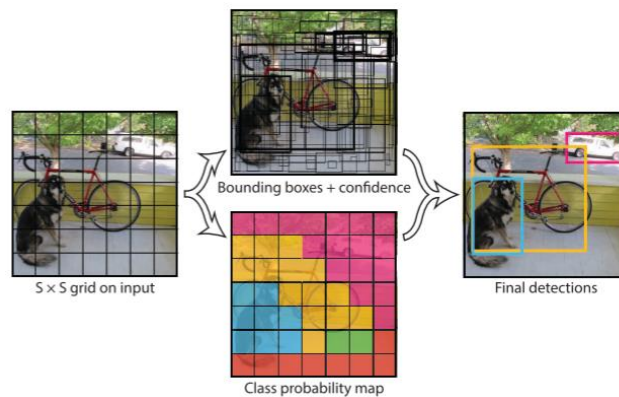
Lapisan ini memiliki tanggung jawab untuk mengklasifikasi fitur yang telah diekstrak. Biasanya, output dari lapisan ini adalah probabilitas untuk setiap kelas dalam tugas klasifikasi.



Gambar 2.1 Arsitektur *Convolutional Neural Network*
(Sumber: <https://www.trivusi.web.id/2022/04/algoritma-cnn.html>)

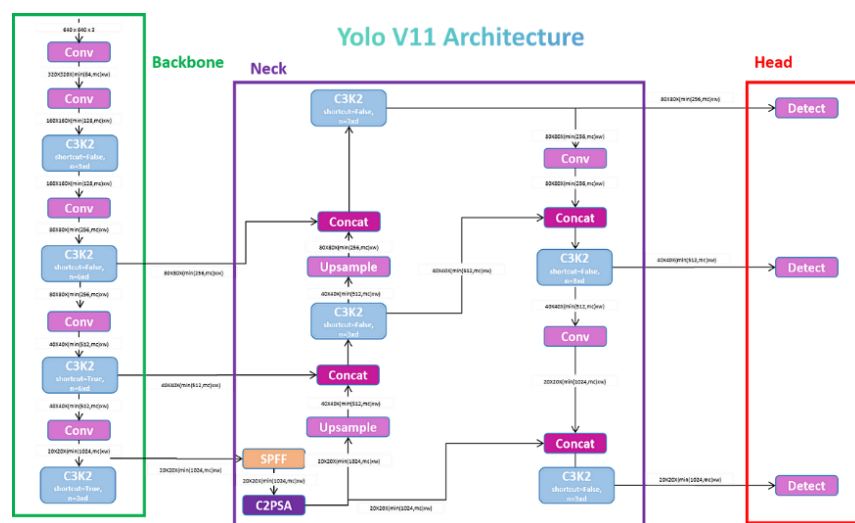
2.1.6 YOLO (*You Only Look Once*)

YOLO adalah salah satu metode deteksi objek berbasis *deep learning* yang populer karena kecepatannya dalam memproses pendeteksian objek dan juga secara *real-time*. YOLO bekerja dengan cara membagi gambar menjadi beberapa bagian *grid* (kotak sel) yang setiap *grid* bertanggung jawab untuk mendeteksi objek lalu dilanjutkan dengan melakukan prediksi *bounding box* dan *confidence score* serta memprediksi kelas dalam satu tahap di dalam satu tahapan seperti pada gambar 2.2 (Redmon dkk., 2016). Hal ini membuatnya lebih cepat dibandingkan metode deteksi objek berbasis *region proposal network* (jaringan konvolusional penuh yang secara bersamaan memprediksi batas objek dan skor objektivitas) seperti R-CNN.



Gambar 2.2 Tahapan Cara Kerja YOLO
(Redmon dkk., 2016)

Sebelum melakukan tahapan pembagian *grid*, Gambar akan melewati proses ekstraksi fitur menggunakan *Convolutional Neural Network* (CNN), tujuannya adalah menghasilkan *feature map* (representasi numerik dari ciri-ciri penting gambar). YOLO memiliki banyak versi dan setiap versi merupakan perkembangan arsitektur dari versi sebelumnya. Pada saat ini, versi terbaru dari YOLO adalah YOLOv11 yang arsitekturnya seperti pada gambar berikut.



Gambar 2.3 Arsitektur YOLOv11
(Sumber : <https://www.analyticsvidhya.com/blog/2024/10/yolov11-object-detection/>)

Dalam arsitekturnya, YOLOv11 dibagi menjadi 3 bagian besar yaitu:

1. Backbone

Bagian *backbone* bertanggung jawab untuk mengekstraksi fitur-fitur penting dari gambar *input*. *Backbone* dapat menangkap pola dan detail penting dengan memanfaatkan blok konvolusional dan blok *bottleneck*. Blok C3K2

merupakan blok *bottleneck* yang merupakan blok konvolusional modern dengan parameter *shortcut* yang menentukan apakah akan ada bagian residual atau tidak. Blok C3K2 ini mengoptimalkan aliran informasi di jaringan dengan membagi *feature map* menjadi bagian-bagian kecil, lalu memprosesnya menggunakan konvolusi kernel kecil (3x3) yang lebih cepat dan ringan (Sulake, 2024).

2. Neck

Pada bagian ini, dilakukan pengolahan lanjutan dengan penggabungan untuk memperkaya informasi fitur. Blok SPFF bertugas untuk menggabungkan informasi dari berbagai skala menggunakan *pooling* dengan ukuran kernel berbeda, sehingga memperbaiki deteksi objek kecil tanpa mengorbankan kecepatan. Sedangkan blok C2PSA membantu model lebih fokus pada area penting pada gambar dengan menggunakan mekanisme *attention* berbasis spasial (Sulake, 2024).

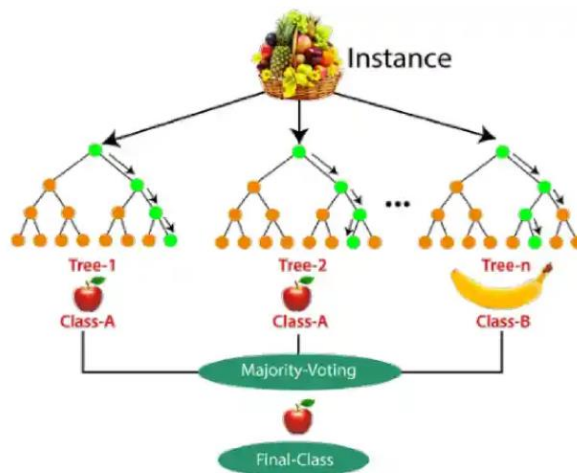
3. Head

Di bagian *head*, dilakukan prediksi multi-skala untuk mendeteksi objek dengan berbagai ukuran. Bagian *head* ini akan menghasilkan *output* deteksi dari tiga *feature map* berbeda (P3, P4, dan P5), yang masing-masing mewakili tingkat kedetailan berbeda dalam gambar. Objek yang kecil akan terdeteksi dengan lebih rinci di skala lebih halus (P3), sedangkan objek besar ditangkap oleh fitur tingkat tinggi (P5) (Sulake, 2024).

YOLO banyak digunakan dalam berbagai aplikasi seperti pengawasan keamanan, deteksi kendaraan dalam sistem transportasi cerdas, dan analisis gambar medis. Model ini dapat menawarkan kecepatan tinggi dan tetap mempertahankan akurasi yang baik dalam mendeteksi objek. YOLO selalu berinovasi dan memperbarui versi modelnya dan selalu mengalami peningkatan dalam efisiensi komputasi, menjadikannya lebih cocok untuk implementasi dalam berbagai sistem. Selain itu, YOLO juga memiliki beberapa varian seperti YOLO-Nano yang dirancang agar dapat berjalan secara efisien pada perangkat *edge* dengan sumber daya terbatas seperti pada ponsel pintar atau perangkat *IoT*.

2.1.7 Random Forest Regressor

Random Forest Regressor adalah salah satu algoritma *machine learning* berbasis *ensemble learning* (teknik yang menggabungkan dua atau lebih pembelajaran) yang digunakan untuk tugas regresi atau memprediksi nilai kontinu berdasarkan satu atau lebih variabel independen. Algoritma ini merupakan pengembangan dari metode *decision tree*, di mana algoritma ini akan membangun sekelompok pohon keputusan (*decision trees*) yang acak, kemudian menggabungkan prediksi dari pohon-pohon tersebut untuk menghasilkan prediksi akhir yang lebih akurat dan stabil (Farhanuddin dkk., 2024).



Gambar 2.4 Arsitektur *Random Forest Regressor*

(Sumber : <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>)

Gambar di atas menunjukkan contoh arsitektur dari *Random Forest Regressor*. Jumlah sampel ($n_{estimator}$) akan diambil dari keranjang buah, setiap sample akan dibangun pohon keputusan individual untuk menentukan *class*. *Output* akhir akan berdasarkan suara mayoritas, dari gambar tersebut terdapat 3 pohon keputusan yang 2 diantaranya menghasilkan *class* Apel sebagai *output*. Dengan begitu maka hasil *output* akhir adalah apel karena mayoritas pohon keputusan menghasilkan apel.

2.1.8 Roboflow

Roboflow adalah platform yang memudahkan pengelolaan *dataset* secara komprehensif untuk proyek yang berhubungan dengan *computer vision* (Saepudin dkk., 2024). Dengan menggunakan Roboflow, pengguna dapat menyiapkan *dataset*

dalam format yang kompatibel dengan berbagai model *deep learning* seperti YOLO, TensorFlow, dan PyTorch. Dengan menyediakan fitur otomatisasi dalam *preprocessing* data seperti normalisasi, *resizing*, dan augmentasi gambar, *Roboflow* dapat membantu meningkatkan kinerja model deteksi objek dengan lebih mudah. Platform ini sering digunakan oleh pengembang dan peneliti untuk mempercepat proses pengolahan data terutama deteksi objek dan segmentasi gambar.

2.1.9 Python

Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 (Alfarizi dkk., 2023). Bahasa ini digunakan secara luas dalam berbagai bidang seperti pengembangan web, *data science*, kecerdasan buatan, dan *machine learning*. Salah satu keunggulan utama Python adalah dukungan pustaka yang sangat luas seperti TensorFlow dan PyTorch untuk *deep learning* atau Pandas dan NumPy untuk analisis data. Dengan komunitas yang besar dan ekosistem yang berkembang pesat, Python menjadi pilihan utama bagi pengembang dalam mengembangkan solusi berbasis kecerdasan buatan dan *computer vision*.

2.1.10 TensorFlow Lite

TensorFlow Lite adalah *library* yang mengoptimalkan model *machine learning* agar dapat dijalankan pada perangkat dengan sumber daya yang terbatas seperti ponsel pintar (Imanuel & Fenriana, 2024). TensorFlow Lite ini dirancang agar model *deep learning* dapat dieksekusi secara efisien tanpa harus mengandalkan server atau komputasi berbasis *cloud*. Hal ini dapat dilakukan karena TensorFlow Lite menggunakan fitur optimasi seperti kuantisasi dan akselerasi perangkat keras. Dengan begitu, TensorFlow Lite menjadi solusi yang efisien untuk berbagai kebutuhan *machine learning* dalam perangkat *mobile* dan *embedded system*.

2.1.11 Android

Android adalah sistem operasi berbasis Linux yang dirancang khusus untuk perangkat seluler seperti ponsel pintar dan tablet. Dikembangkan oleh Google, Android menyediakan platform terbuka yang memungkinkan pengembangan untuk membuat berbagai aplikasi yang dapat berjalan di berbagai perangkat. Keunggulan

utama *Android* adalah fleksibilitasnya dalam mendukung berbagai perangkat keras, ekosistem aplikasi yang luas, dan kemampuan untuk dikustomisasi sesuai kebutuhan pengembang dan produsen perangkat. Menurut data dari StatCounter, *Android* menempati peringkat pertama menjadi sistem operasi *mobile* yang paling banyak digunakan di dunia dengan pangsa pasar global mencapai 71.9% dibulan Maret 2025 (StatCounter, 2025).

2.1.12 Visual Studio Code

Visual studio Code adalah *code editor* yang efisien dengan yang digunakan untuk mengembangkan aplikasi dengan berbagai fitur lainnya seperti *debugging*, menjalankan tugas, dan kontrol versi. *Code editor* ini mendukung banyak bahasa pemrograman dengan *build-in* seperti JavaScript, Python, C++, dan lainnya (Hartati, 2020). Selain itu, Visual Studio Code memiliki ekstensi yang luas sehingga memungkinkan menyesuaikan lingkungan kerja sesuai kebutuhannya. Dengan adanya fitur integrasi Git, Visual Studio Code dapat memudahkan pengelolaan versi kode langsung dari editor. Dengan begitu, Visual Studio Code menjadi pilihan utama bagi banyak pengembang dalam membangun berbagai jenis aplikasi.

2.1.13 Flutter

Flutter adalah kerangka kerja *open-source* untuk pengembangan aplikasi yang dapat berjalan dalam sistem operasi *Android* atau *ios* (Krisnada & Tanone, 2019). Diperkenalkan oleh Google pada Mei 2017, Flutter telah berkembang pesat dan menjadi salah satu kerangka kerja yang paling banyak digunakan oleh developer. Flutter memiliki beberapa kelebihan yang membuatnya lebih dipilih oleh developer seperti lebih mudah mengembangkan aplikasi, *user interface* yang dapat di custom, performa yang optimal, dan yang pasti multiplatform.

2.1.14 MAP (Mean Average Precision)

MAP adalah metrik evaluasi utama dalam *object detection* yang mengukur performa model dalam mendeteksi dan mengklasifikasikan objek pada gambar (Wang, 2022). Nilai mAP diperoleh dari rata-rata presisi di berbagai nilai *recall*, yang memberikan gambaran tentang sejauh mana model dapat mengenali objek dengan akurat.

Perhitungan mAP melibatkan konsep *Precision-Recall Curve*. Presisi mengukur seberapa banyak prediksi yang benar dibandingkan dengan sebuah prediksi yang dibuat, sementara *recall* mengukur seberapa banyak objek yang berhasil dideteksi dibandingkan dengan total objek yang ada. Nilai mAP dihitung dengan mengambil rata-rata dari *Average Precision* (AP) pada berbagai nilai *threshold Intersection over Union* (IoU). Semakin tinggi nilai mAP yang didapat ketika mengevaluasi model pendeteksi objek, maka mengartikan semakin baik performa dari model tersebut. Rumus umum untuk mAP adalah sebagai berikut :

$$MAP = \frac{1}{n} * \sum_{i=1}^n AP_i$$

Yang dimana :

- AP_i adalah *Average Precision* untuk setiap kelas
- n adalah jumlah kelas dalam *dataset*

2.1.15 MAE (Mean Absolute Error)

MAE merupakan salah satu metode yang digunakan untuk mengevaluasi hasil prediksi dari model pembelajaran mesin (*machine learning*) dengan cara mencari nilai rata-rata absolut dari kesalahan prediksi (Gusdian dkk., 2016). Adapun rumus dalam perhitungan mAE adalah sebagai berikut :

$$MAE = \frac{1}{n} * \sum_{i=1}^n |y_i - \hat{y}_i|$$

Yang dimana :

- y_i adalah nilai aktual
- $\hat{y}_i$ adalah nilai prediksi
- n adalah jumlah data

Dari hasil skor MAE ini, karena merupakan perhitungan kesalahan prediksi maka model akan semakin baik kualitasnya apabila skor MAE semakin kecil.

2.1.16 Black Box Testing

Black box adalah istilah yang digunakan untuk menggambarkan sistem yang kompleks dan sulit dipahami secara mendalam. *Black box testing* merupakan salah satu metode pengujian pada perangkat lunak (*software*) dengan memeriksa nilai keluaran berdasarkan nilai masukan saja tanpa harus memeriksa detail seperti kode

program dalam perangkat lunak tersebut (Ningrum dkk., 2019). Tujuan dilakukannya *black box testing* adalah untuk menemukan fungsi yang tidak benar sehingga dapat dilakukan evaluasi dan perbaikan agar mendapatkan hasil yang sesuai dengan keinginan.

2.2 Penelitian Terdahulu

Penelitian terdahulu yang dilakukan Imantiyar (2021), membuat sebuah pengujian tentang pengaruh bias *dataset* terhadap performa akurasi pendeteksian objek. Penelitian ini menggunakan model *efficientdet-lite* untuk mendeteksi barang dengan alasan akurasi yang baik dan kinerja yang optimal di perangkat *mobile*. Dalam pengujiannya, model dilatih dengan 4 *dataset* yang berbeda yaitu *dataset* optimal, *dataset* dengan bias arah datangnya cahaya, *dataset* dengan bias sudut penangkapan gambar, dan bias pada label. Jika kita ambil model dengan *dataset* optimal yang mendapat nilai AP (*Average Precision*) sebagai acuan, hasil dari pengujian tersebut mengatakan bahwa bias cahaya mengalami penurunan nilai AP sebesar 36.9%, bias pengambilan sudut mengalami penurunan nilai AP sebesar 16%, dan yang paling fatal adalah *dataset* dengan bias labeling yang mengalami penurunan hingga 98% jika dibandingkan dengan model *dataset* optimal.

Penelitian yang dilakukan oleh Riyazudin dkk. (2022), memperkenalkan pendekatan berbasis *deep learning* yang digunakan untuk mendeteksi makanan dan menghitung kalori menggunakan foto makanan. Untuk menghitung kalori, penelitian tersebut menggunakan objek referensi perbandingan rasio ukuran dengan bahan makanan. Objek referensi akan diinputkan pengguna dan juga harus menetapkan tinggi atau lebarnya. Untuk inputannya, akan ada 2 gambar yakni gambar tampak atas dan tampak depan. Dalam penelitiannya, model yang digunakan untuk pendeteksian objek adalah YOLOv3. Dalam evaluasinya, akan menggunakan mIoU (*Mean Intersection over Union*) yang merupakan matriks dalam tugas segmentasi pada *computer vision*. Pada penelitian tersebut, Model penelitian ini mendapatkan nilai mIoU segmentasi di angka 0.800. Hasil dari penelitian tersebut adalah sebuah situs web.

Penelitian selanjutnya adalah penelitian dari Carrión dkk. (2022), dengan judul penelitian deteksi luka otomatis dan estimasi ukuran menggunakan *deep learning*. Kajian dari penelitian ini yang akan diambil adalah bagian estimasi

ukuran pada luka yang terdeteksi. Dalam penelitian ini, peneliti menggunakan objek referensi yaitu belat berbentuk cincin di sekitar luka dengan ukuran yang diketahui. Dengan adanya objek referensi ini, model yang dilatih dapat mengestimasi ukuran dari luka.

Penelitian yang dilakukan oleh Nursulistio dkk. (2022) membuat pendeteksian objek masker yang memanfaatkan *transfer learning*. Pada penelitiannya model akan dibuat dapat mendeteksi jenis masker yang digunakan dan juga mendeteksi penggunaan masker yang salah ataupun tanpa masker. Untuk model yang digunakan adalah *efficientdet-lite3* yang merupakan salah satu *pretrained* model yang disediakan tensorflow. Dalam penelitiannya, dibuat 2 skenario dalam melakukan *retrain* model, yang pertama menggunakan *dataset* bias (ketidakseimbangan dalam *dataset*) dan yang kedua menggunakan *dataset* yang sudah disamaratakan. Hasilnya membuat model mendapatkan nilai rata-rata presisi sebesar 75.84% pada skenario kedua yang lebih tinggi 0.02% dibanding skenario pertama.

Penelitian yang dilakukan Riswanto dkk. (2024) membuat pengembangan sistem pendeteksi kalori makanan tradisional Indonesia. Untuk model yang digunakan adalah SSD (*Single Shot Multibox Detector*) yang akan dilatih dengan *dataset* makanan tradisional Indonesia. Dalam evaluasi model menggunakan data uji, model ini mendapatkan nilai mAP (*mean average precision*) sebesar 65.09% .

Penelitian yang akan dilakukan dalam proyek akhir ini akan membangun mengembangkan sebuah sistem pendeteksi kalori pada makanan dengan memanfaatkan *deep learning* dan *machine learning*. Sistem yang dibangun akan diimplementasikan ke dalam bentuk aplikasi *mobile* berbasis *Android*. Model *deep learning* yang akan digunakan untuk melakukan pendeteksian makanan adalah YOLOv11n dan model *machine learning* yang digunakan dalam pengestimasian berat adalah *Random Forest Regressor*. Pengestimasian berat dilakukan berdasarkan ukuran dari makanan yang akan menggunakan sebuah piring sebagai objek referensi untuk mendapatkan rasio.

Tabel 2.1 Perbandingan penelitian terdahulu

No	Peneliti	Judul Penelitian	Algoritma	Hasil
1	(Imantiyar, 2021)	Pengaruh Bias Dataset Terhadap Performa Akurasi	EfficientDet-Lite	Pengujian seberapa pentingnya meminimalisir bias (ketidak akuratan data)

No	Peneliti	Judul Penelitian	Algoritma	Hasil
		Deteksi Objek Pada Arsitektur EfficientDet-Lite		pada <i>dataset</i> untuk meningkatkan akurasi pada arsitektur yang digunakan.
2	(Riyazudin dkk., 2022)	Estimation of Quantity and Nutritional Information of Food Using Image Processing	YOLO v3	Web pengestimasi kalori bahan makanan yang memanfaatkan objek referensi yang diinputkan pengguna lalu berukuran tetap untuk mengestimasi berdasarkan ukuran ukuran.
3	(Carrión dkk., 2022)	Automatic wound detection and size estimation using deep learning algorithms	YOLOv3 dan U-Net	Model pendeteksi luka dengan menggunakan sebuah objek referensi sebagai cincin untuk dibandingkan dengan ukuran luka sehingga dapat mengestimasi ukuran dari luka
4	(Nursulistio dkk., 2022)	Deteksi Objek Masker Menggunakan EfficientDet-Lite3	EfficientDet-Lite3	Model pendeteksian penggunaan masker dengan memanfaatkan transfer learning
5	(Riswanto dkk., 2024)	Deteksi Kalori Makanan Tradisional Indonesia Menggunakan Metode Single Shot Multibox Detector (SSD)	Single Shot Multibox Detector	Aplikasi mobile yang dapat mendeteksi makanan dengan nilai mAP 65.09%
6	Renggodhani	Perhitungan Kalori Makanan Dengan Menggunakan YOLOv11 dan Random Forest Regressor	YOLOv11n dan Random Forest Regressor	