

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [8] pengujian menggunakan algoritma *YOLO* dan *CNN*. Dapat Mengidentifikasi dan mengklasifikasikan berbagai jenis kendaraan. Terdapat 16.185 foto mobil dalam Stanford Car Dataset, yang digunakan untuk tujuan pengujian di bidang penandaan kendaraan. Dipisahkan menjadi 8.144 foto pelatihan dan 8.041 gambar pengujian, dataset ini memiliki 196 varian kendaraan yang berbeda. Ada seratus kelas dalam set pelatihan Stanford Car Dataset, dan lima puluh foto membentuk setiap kelas. Di sisi lain, dataset pengujian memiliki 100 kelas, dan untuk setiap kelas terdapat 10 foto. Keakuratan model yang dilatih dengan dataset tersebut dievaluasi dengan tes deteksi dan pengenalan. Dalam percobaan ini, kami memilih secara acak 100 kelas, dimulai dengan masing-masing 10 foto, dari dataset Stanford Car Image.

Penelitian [9] YOLOv3 mampu mendeteksi penggunaan masker secara real-time tanpa dipengaruhi oleh kondisi cahaya, sudut objek, atau atribut yang diterapkan dengan baik. Tingkat keakuratan atau nilai kepercayaan yang diperoleh dari berbagai situasi mencapai 71-99% saat objek tidak menutupi area wajah dengan menggunakan hijab sebagai pengganti masker, dengan objek berjarak antara ± 2 hingga ± 5 meter dari kamera. Namun, dalam situasi di mana objek menutupi area wajah dengan menggunakan hijab sebagai pengganti masker, kepercayaan deteksi hanya sebesar 0.28 atau 28% dengan klasifikasi yang benar sebagai tidak menggunakan masker. Hal ini disebabkan oleh kekurangan data pelatihan pada sistem untuk mengatasi kondisi seperti itu.

Penelitian [10] penggunaan pemetaan *bounding box* pada berbagai jenis objek dalam citra telah berhasil terlihat melalui analisis deteksi. Hasil yang menunjukkan tingkat akurasi yang tinggi mengindikasikan bahwa penggunaan YOLOv4 dalam mendeteksi objek berjalan secara efisien dan

sesuai dengan anotasi data citra yang ada. Implementasi deteksi objek ini memiliki potensi besar dalam penggunaannya untuk mengidentifikasi korban dalam situasi bencana saat proses evakuasi, yang dapat secara signifikan mengurangi waktu pencarian dan mengurangi jumlah korban karena telah teridentifikasi melalui sistem deteksi YOLOv4.

Penelitian [11] pengujian kesegaran ikan menggunakan algoritma YOLOv4-CNN, hasil menunjukkan kinerja yang efektif. Penggunaan metode CNN untuk mengklasifikasikan ikan sebagai segar atau tidak segar terbukti berhasil. Hasil klasifikasi ini kemudian diaplikasikan kembali pada YOLOv4 untuk menilai deteksi ikan segar secara optimal berdasarkan model yang telah ditetapkan. Perlu dicatat bahwa dalam proses deteksi, terdapat variasi nilai *confidence* pada setiap *frame* karena pergerakan dan perubahan posisi objek. Dengan penggunaan 118 data latih dan 13 data uji serta pelatihan selama 5000 *epoch*, sistem berhasil mendeteksi kesegaran ikan dengan baik. Uji coba YOLOv4-CNN menghasilkan akurasi *mAP* sebesar 93.75%, dengan presisi 1.00%, *recall* 0.93%, *fScore* 0.96%, dan rata-rata nilai IoU sebesar 74.17%. Penelitian ini menunjukkan hasil yang sangat baik dalam menguji kinerja algoritma tersebut.

Penelitian [12] hasil pengujian pada sistem dan aplikasi, YOLOv3 yang telah diperlakukan dengan pelatihan lanjutan untuk mengenali lokasi pakaian, mencapai tingkat akurasi sebesar 82.5%. Hal ini dicapai dengan menggunakan dataset yang telah diambil dari dataset yang lebih besar. Namun, YOLOv3 masih memiliki kelemahan dalam mendeteksi semua objek pakaian dalam gambar. Kadang-kadang, kemampuannya hanya terbatas pada satu atau dua objek saja. Sementara itu, klasifikasi ResNet50 yang telah disesuaikan (*fine-tuning*) dengan dataset uji mampu memberikan hasil yang memuaskan, dengan tingkat akurasi sebesar 86.44%, presisi 86.05%, *recall* 86.16%, dan *f-score* 86.11%. Namun, pengujian terhadap model ResNet50 yang telah disesuaikan hanya mampu menerima gambar-gambar yang memiliki tingkat blur pada skala PIL antara 1 hingga 4, yang termasuk dalam kategori blur ringan.

Penelitian [13] deteksi objek di wilayah kepala melalui penerapan CNN menunjukkan kinerja unggul, seperti yang terbukti melalui serangkaian uji coba yang dilakukan pada pengenalan objek tanpa helm dalam kondisi yang mungkin mempengaruhi hasil deteksi helm dan tanpa helm. Keakuratan deteksi helm dengan menggunakan dataset pelatihan, sebagai metode evaluasi model, mencapai tingkat akurasi rata-rata sebesar 89.04%, sementara nilai rata-rata kehilangan pelatihan (*avg_loss*) mencapai 1.2%.

Penelitian [14] mengusulkan sistem untuk mendeteksi kematangan pisang menggunakan YOLOv4. Berdasarkan hasil yang diperoleh pada tahap pengujian, tingkat akurasi rata-rata terbaik yang diperoleh adalah 87,6%. Hasil ini diperoleh dengan menggunakan bobot pada iterasi ke-5000 (dari maksimal 6000 iterasi) pada fase pelatihan. Untuk pengujian deteksi kematangan secara *real-time* menggunakan webcam, perangkat menghasilkan *frame per second* (FPS) sebanyak 5 FPS. Sebagai masukan untuk penelitian lebih lanjut, diperlukan implementasi arsitektur multiGPU untuk meningkatkan rata-rata FPS selama deteksi secara *real-time*.

Penelitian [15] implementasi deteksi Sarang Orang Utan menggunakan metode YOLO dilakukan dengan menginputkan 1970 citra gambar, termasuk 414 gambar berlabel Sarang Orang Utan berukuran 1920x1080 pixel. Dalam skenario perbandingan dataset, 79% digunakan untuk pelatihan dan 21% untuk pengujian. Pengujian ulang menggunakan 10 citra test menunjukkan bahwa model, berdasarkan *training weight* terbaik dengan menggunakan PyTorch, mampu mendeteksi jumlah sarang orang utan dengan rata-rata akurasi mencapai 99,9%.

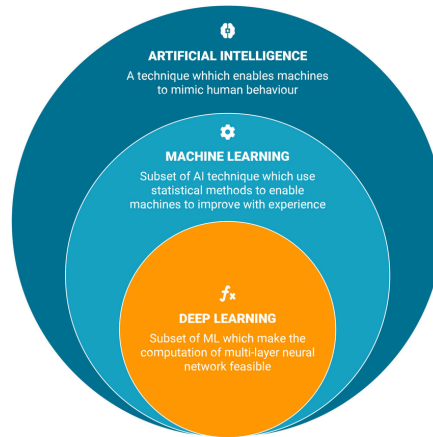
Penelitian [16] algoritma YOLO berhasil mengenali objek kebakaran dalam video dengan tingkat keakuratan yang memuaskan, diperoleh nilai kepercayaan rata-rata sebesar 0.66. Rentang nilai *confidence* berkisar antara 0.55 hingga 0.71. Dengan menetapkan *threshold* sebesar 0.30, hasil deteksinya menunjukkan skor *precision* mencapai 0.98, *recall* sebesar 0.95, dan akurasi sebesar 0.95. Selain itu, skor *Mean Average Precision (mAP)* yang dihasilkan mencapai 72.63%.

2.2 DASAR TEORI

2.2.1 *Artificial Intelligence (AI)*

Bidang ilmu komputer yang dikenal sebagai "*Artificial Intelligence*" berupaya menciptakan program dan sistem komputer yang dapat meniru kecerdasan manusia. Algoritme dan model matematika adalah blok bangunan kecerdasan buatan, yang memungkinkan mesin untuk memahami data, menemukan pola, dan membuat pilihan cerdas. Pembelajaran mesin, jaringan saraf, NLP, dan banyak lagi bidang lainnya merupakan hal yang penting bagi kecerdasan buatan. Banyak bidang yang sangat terpengaruh oleh kemajuan AI, termasuk perawatan kesehatan, mobil otonom, pengenalan suara, pengenalan wajah, dan masih banyak lagi [17].

AI merujuk pada suatu ranah dalam ilmu komputer yang bertujuan untuk meningkatkan kapabilitas mesin atau komputer agar mampu menyelesaikan tugas-tugas serupa manusia atau bahkan melampaui kemampuan manusia. Dalam praktiknya, AI memanfaatkan pengetahuan khusus untuk mengalihkan tugas-tugas manusia ke dalam lingkungan komputer dengan maksud untuk memecahkan masalah atau memberikan saran. Perkembangan kecerdasan buatan diperkirakan akan mengubah cara hidup kita lebih cepat daripada yang pernah diantisipasi sebelumnya. Kecerdasan buatan mencakup kemampuan mesin untuk mempelajari informasi dari data melalui algoritma, dan kemudian menggunakan pengetahuan yang diperoleh untuk membuat keputusan mirip manusia. AI bukan hanya sekadar sistem yang dapat berpikir dan berperilaku seperti manusia, tetapi juga merupakan suatu sistem pemikiran rasional dan fungsional [18].



Gambar 2.1 Bagian Artificial Intelligence [18]

AI melibatkan beragam disiplin ilmu, mencakup di antaranya *machine learning* yang mencantumkan *deep learning* sebagai fokus penelitian dalam situasi ini. Berdasarkan Gambar 2.1, terlihat bahwa terdapat keterkaitan antara *AI*, *machine learning*, dan *deep learning*.

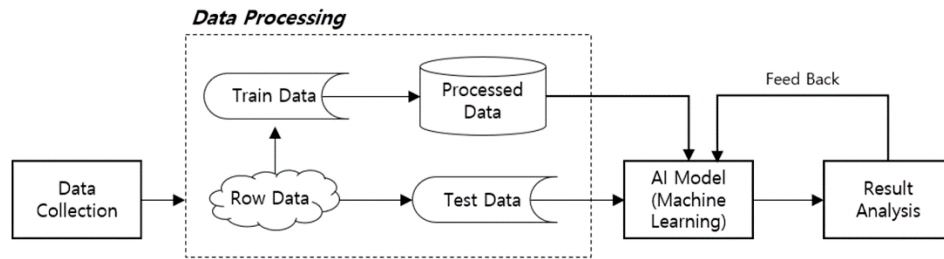
AI memiliki beberapa kelebihan yang membuatnya menjadi teknologi yang sangat menjanjikan di berbagai bidang. Kelebihan *AI* sebagai berikut :

1. Tahan lama, tidak seperti kecerdasan alami, yang dapat berubah seiring waktu karena orang melupakan sesuatu. Dalam hal komputer dan program-programnya, kecerdasan buatan, di sisi lain, bersifat statis.
2. Sangat mudah untuk menyalin dan mendistribusikan kecerdasan buatan. Butuh waktu lama untuk mewariskan pengetahuan manusia, dan tidak selalu memungkinkan untuk meniru pekerjaan seorang ahli sepenuhnya. Akibatnya, informasi yang disimpan dalam sistem komputer dapat dengan mudah dipindahkan ke komputer lain.
3. Dibandingkan dengan melibatkan manusia untuk melakukan sejumlah pekerjaan dalam jangka waktu yang lama, kecerdasan buatan lebih hemat biaya karena menyediakan layanan komputer dengan biaya yang lebih rendah.
4. *AI* bekerja lebih cepat dan lebih efektif daripada metode tradisional [19].

Ada beberapa kelemahan AI yang harus diatasi meskipun ada banyak manfaatnya :

1. Data yang digunakan untuk melatih AI memiliki dampak yang signifikan pada seberapa baik kinerjanya. Sistem AI dapat memberikan temuan yang tidak adil atau salah jika data yang digunakannya bias atau tidak representatif.
2. Pengumpulan dan Analisis Data Pribadi Berpotensi Mengancam Privasi Individu. Sebagai contoh, algoritma kecerdasan buatan yang diterapkan pada analisis perilaku online mampu menghimpun dan menggabungkan data dari berbagai sumber, termasuk media sosial, riwayat penelusuran, dan transaksi online. Dalam konteks ini, informasi terkait preferensi, minat, kebiasaan, atau bahkan data pribadi seperti alamat atau nomor telepon dapat terungkap tanpa disengaja melalui analisis pola yang dilakukan oleh sistem kecerdasan buatan.
3. Salah satu bidang yang rentan terhadap penggantian oleh kecerdasan buatan adalah sektor pekerjaan yang melibatkan tugas-tugas rutin dan terstruktur. Contoh dari sektor ini mencakup pekerjaan administratif, seperti pengolahan data dan dokumen, serta pekerjaan produksi yang melibatkan operasi mesin yang bersifat repetitif. Pemanfaatan robotika dan otomatisasi yang didukung oleh AI dapat menggantikan tugas-tugas tersebut dengan lebih cepat dan efisien, sehingga mengurangi kebutuhan terhadap keterlibatan manusia[20].

Berbagai jenis teknologi kecerdasan buatan (AI), termasuk *deep learning*, terdiri dari analisis dan algoritma pembelajaran, sistem komputasi, dan pembelajaran data yang secara langsung terkait dengan kompleksitas algoritma. Untuk pembelajaran AI, diperlukan contoh data, dan untuk mengembangkan model AI dengan fitur tertentu, dataset yang sesuai harus dibentuk untuk pembelajaran. Alur pembelajaran mesin AI seperti yang ditunjukkan dalam Gambar 2.2.



Gambar 2.2 Blok Diagram Artificial Intelligence [21]

Data Collection : tahap ini bertujuan untuk menghimpun data yang bersifat tidak terstruktur, melibatkan berbagai jenis informasi seperti gambar, teks, dan suara. Pada tahap awal ini, dilakukan pra-pemrosesan yang melibatkan ekstraksi data yang sesuai dengan tujuan dan fungsi AI yang sedang dikembangkan.

Data Preprocessing : Pada tahap ini, data yang telah terkumpul mengalami transformasi agar dapat dimasukkan ke dalam model pembelajaran mesin. Proses ini melibatkan pengisian atau penghapusan nilai yang hilang, pemilihan atau penghapusan properti data, penggabungan properti data yang sudah ada, dan konversi data mentah menjadi format yang telah ditentukan sesuai kebutuhan.

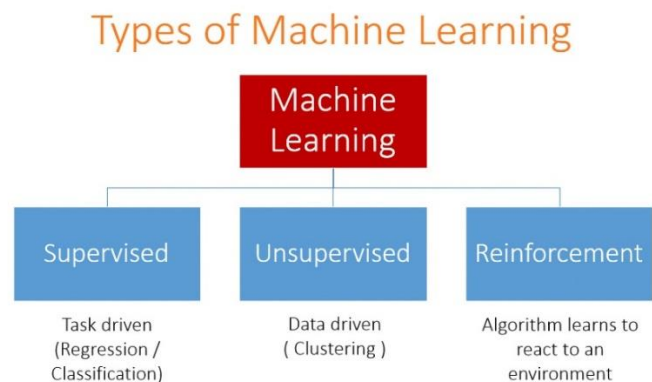
Data Analysis : Pada tahap ini, dilakukan analisis terhadap data yang akan diterapkan pada AI. Langkah-langkah ini mencakup eksplorasi pola data standar, pemetaan data, ekstraksi data berdasarkan eksplorasi dan inferensi, serta pembelajaran data menggunakan sebagian dari data yang ada.

Data awal yang terkumpul selama proses pengumpulan data sering kali *noise*, tidak konsisten, dan mengalami pengulangan, sehingga tidak sesuai untuk diterapkan dalam algoritma kecerdasan buatan. Untuk menjamin kualitas, keandalan, akurasi, dan kinerja yang optimal, diperlukan suatu tahap untuk meningkatkan kualitas data; pada tahap ini, analisis dan organisasi data dilakukan dengan tingkat profesionalisme dan wawasan yang tinggi. Langkah-langkah ini mencakup modifikasi kesalahan data, eliminasi data yang tumpang tindih, penghapusan data yang tidak konsisten, dan penyelesaian konflik data. Pra-pemrosesan data memiliki kontribusi sekitar

80% dari seluruh proses. Memperoleh jumlah data berkualitas tinggi yang memadai menjadi sangat penting untuk pengembangan AI, dan penjaminan kualitas data pembelajaran AI menjadi suatu keharusan [21].

2.2.2 Machine Learning

Machine Learning dapat dijelaskan sebagai penerapan Komputer dan algoritma matematika yang menggunakan teknik pembelajaran dari data untuk menghasilkan prediksi di masa depan dikenal sebagai pembelajaran mesin. Pelatihan dan pengujian adalah dua komponen utama dari proses pembelajaran ini yang bertujuan untuk meningkatkan kecerdasan. Penelitian terbaru menunjukkan bahwa ada tiga subbidang utama dalam *machine learning*: *Supervised Learning*, *Unsupervised Learning*, *Reinforcement Learning* [22].



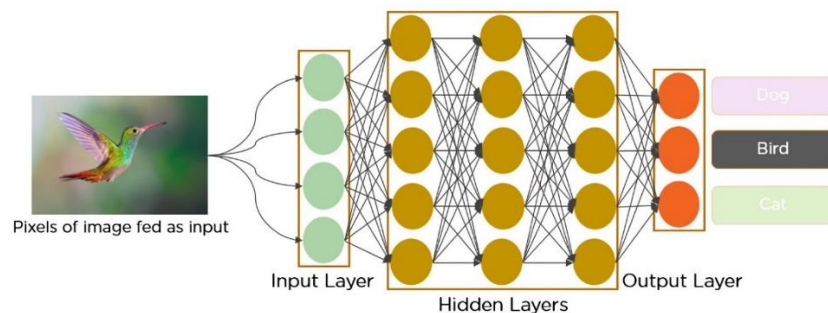
Gambar 2.3 Type of Machine Learning [22]

Pada Gambar 2.3, *Supervised Learning* menggunakan kumpulan data yang telah diberi label untuk mengkategorikan kelas yang tidak diketahui; ini adalah pendekatan klasifikasi. *Unsupervised Learning* sering disebut cluster dikarenakan alah satu jenis pembelajaran tanpa pengawasan; dalam pendekatan ini, pengumpulan data tidak diberi label dan hasilnya tidak digunakan untuk mengklasifikasikan contoh ke dalam kategori yang telah ditentukan. *Reinforcement Learning* umumnya, teknik ini beroperasi di antara *Supervised Learning* dan *Unsupervised Learning*. Metode ini berfungsi sebagai jembatan antara pembelajaran yang diawasi dan tidak diawasi. Pendekatan ini bekerja dalam lingkungan yang dinamis ketika idenya adalah

untuk mencapai tujuan tanpa menerima umpan balik langsung dari komputer ketika tujuan tersebut terpenuhi [23].

2.2.3 Deep Learning

Deep Learning merupakan cabang dari *Machine Learning* yang terinspirasi oleh struktur korteks manusia. Dalam implementasinya, *Deep Learning* menggunakan jaringan syaraf buatan yang memiliki sejumlah lapisan tersembunyi. Salah satu metode yang populer dalam *Deep Learning* adalah CNN, yang dirancang untuk mengatasi kelemahan yang ada pada metode sebelumnya. Beberapa kekurangan pada metode sebelumnya dapat diatasi dengan menggunakan model ini, termasuk pengurangan jumlah parameter bebas dan penanganan deformasi gambar input seperti translasi, rotasi, dan skala [24].



Gambar 2.4 Layer pada *deep learning* [24]

Pada Gambar 2.4, struktur dalam *deep learning* terdiri dari tiga komponen utama, yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* mengandung node-node yang menyimpan nilai masukan yang tetap selama fase latihan dan berubah hanya saat menerima masukan baru, pada penelitian ini input yang dimaksud adalah seluruh sampel atau gambar ikan bawal yang terdiri atas ikan bawal bagus, cacat ekor dan cacat sirip. Di *hidden layer*, gambar ketiga ikan di klasifikasikan untuk belajar membedakan pada masing-masing kelasnya. Struktur lapisan dapat membentuk kombinasi algoritma yang optimal untuk mengurangi kesalahan pada keluaran. *Output layer* menggunakan fungsi aktivasi untuk menghasilkan hasil perhitungan sistem berdasarkan masukan yang diterima. Tanggung jawab *output layer*

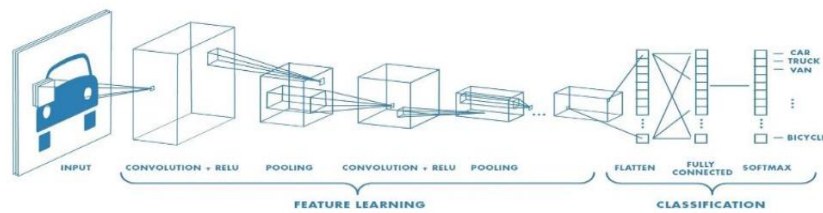
adalah menampilkan hasil akhir dari proses tersebut. Hasil prosesnya yaitu mengetahui gambar ikan di ketiga kelasnya.

2.2.4 Computer Vision

Computer Vision adalah proses di mana gambar atau video dianalisis dan dipahami oleh sistem komputer, menyerupai kemampuan manusia dalam menginterpretasi visual. Dengan kata lain, *computer vision* berupaya meniru cara kita manusia memahami visual. Bidang ini terkait erat dengan pengolahan citra dan visi mesin, di mana terdapat persamaan yang mencolok dalam teknik dan aplikasi yang digunakan. Ini menunjukkan bahwa dasar-dasar teknik yang digunakan dalam ketiga bidang tersebut memiliki kesamaan yang cukup besar. Selain itu, *computer vision* juga terhubung dengan berbagai bidang lainnya seperti kecerdasan buatan, robotika, industri otomatisasi, pengolahan sinyal, optik fisik, neurobiologi, dan sebagainya. Saat ini, *computer vision* banyak digunakan dalam berbagai konteks, misalnya untuk mendeteksi wajah dalam gambar (deteksi wajah), mengenali ekspresi wajah (pengenalan ekspresi wajah), dan sering kali dikombinasikan dengan jaringan syaraf tiruan dalam praktiknya [25].

2.2.5 Convolutional Neural Network (CNN)

Salah satu jenis model saraf tiruan yang dikembangkan untuk menangani masalah pemrosesan gambar adalah *CNN*. *CNN* adalah alat yang efektif untuk pengenalan, klasifikasi, dan identifikasi pola gambar. *CNN* sangat baik dalam memahami detail gambar yang lebih halus karena arsitekturnya meniru cara otak manusia menangani data visual. *CNN* menggunakan matriks dan bobot empat dimensi, yang secara kolektif dikenal sebagai kernel konvolusi, untuk memproses input dua dimensi, seperti gambar atau suara. Jaringan syaraf tiruan *CNN* terbatas pada data dua dimensi karena prinsip konvolusi ini [26].



Gambar 2.5 Arsitektur dari CNN

Pada gambar 2.5, secara simpel dapat diidentifikasi dua bagian utama dalam proses CNN, yakni pembelajaran fitur dan klasifikasi. Dalam segmen pembelajaran fitur, dilakukan tiga tahap esensial, yaitu:

1. Melakukan operasi konvolusi, langkah utama dalam CNN dengan tujuan mengekstrak fitur dari citra masukan.
2. Menggunakan fungsi aktivasi ReLu (*Rectified Linear Unit*), yang menghasilkan output 0 jika input kurang dari 0, artinya $x = 0$ jika $x \leq 0$, dan $x = x$ jika $x > 0$.
3. Melakukan proses *pooling*, yang bertujuan mengurangi parameter pada peta fitur dan mengambil informasi krusial melalui operasi *down-sampling*. Umumnya, metode *pooling* yang diterapkan adalah *max pooling* dan *average pooling*.

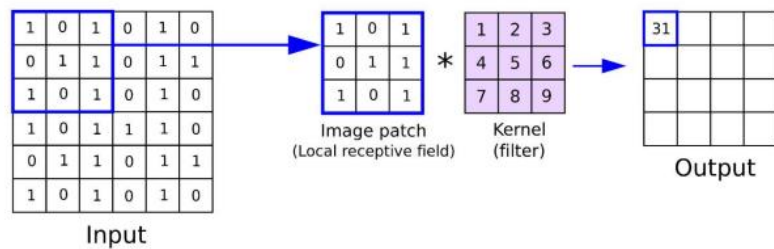
Sementara pada bagian klasifikasi, terdapat tiga langkah, yaitu:

1. Menyusun data menggunakan *flatten layer*, yang mengubah output dari layer konvolusi menjadi satu kolom vektor fitur untuk digunakan pada layer terhubung secara penuh.
2. Menggunakan *fully connected layer*, mirip dengan *Artificial Neural Network* (ANN), untuk melakukan proses klasifikasi.
3. Menerapkan fungsi *softmax*, digunakan untuk menghasilkan nilai probabilitas dari setiap kelas, di mana nilai probabilitas kelas tertinggi menjadi output prediksi kelas. Keseluruhan proses ini merupakan operasi standar yang umum digunakan dalam berbagai model CNN.

Algoritma pada CNN ini sangat sederhana terdiri dari *input layer*, *output layer*, dan *hidden layers*. Dalam *hidden layers* mempunyai beberapa lapisan yaitu ada:

1. Convolution Layer

Convolution layer merupakan bagian penting dari metode CNN, lapisan konvolusi menerapkan operasi konvolusi pada lapisan output sebelumnya. Berikut adalah ilustrasi proses konvolusi pada citra yang ditunjukkan pada gambar 2.6.



Gambar 2.6 Representasi visual layer konvolusi [27]

Pada Gambar 2.6. Input Image digunakan untuk memasukan gambar dengan matriks 6x6 yang terdiri dari angka 0 dan 1. Sebuah potongan kecil dari gambar masukan yang berukuran 3x3 dipilih. Ini disebut sebagai *image patch* atau *local receptive field*. Patch ini berfungsi sebagai bagian kecil dari gambar yang akan dianalisis. Kernel atau filter adalah matriks 3x3 yang memiliki angka-angka 1 hingga 9. Filter ini akan diterapkan pada *image patch* untuk melakukan operasi konvolusi. Nilai hasil konvolusi ditempatkan di lokasi yang sesuai pada gambar output. Dalam contoh ini, satu nilai output (31) dihasilkan dari satu patch konvolusi [27].

2. Pooling Layer

Pooling layer adalah langkah ekstraksi setelah *convolution layers* konsepnya adalah menggunakan filter yang mencakup seluruh wilayah peta fitur dengan ukuran dan langkah tertentu. Tujuan utama dari prosedur pengumpulan ini adalah untuk melakukan pengambilan sampel ke bawah untuk menyederhanakan lapisan berikutnya. Hal ini dapat dilihat sebagai langkah menuju penurunan resolusi gambar atau ukuran matriks dalam konteks pemrosesan gambar. Baik pengumpulan maksimum maupun pengumpulan rata-rata digunakan

secara luas, dengan pengumpulan maksimum mengambil nilai terbesar dan pengumpulan rata-rata mengambil nilai rata-rata [27].

3. *Fully connected layer*

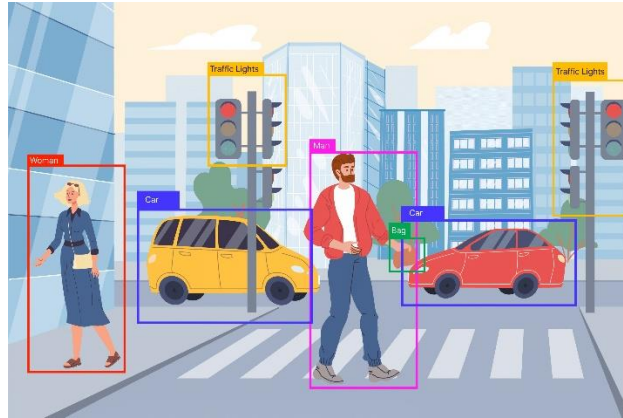
Fully connected layer mengambil input dari hasil output *pooling layer* yang berupa *feature mAP*. Untuk menghasilkan vektor dengan n dimensi-di mana n adalah jumlah kelas keluaran yang akan dipilih oleh program-peta fitur dibentuk ulang dari bentuk larik multidimensi aslinya. Pada lapisan 500 neuron, misalnya, output klasifikasi akhir jaringan akan berupa daftar sepuluh label kelas dengan probabilitas tertinggi, yang diperoleh dengan menerapkan fungsi softmax [28].

4. *Rectified linear unit (ReLU)*

ReLU merupakan elemen dalam jaringan konvolusi yang bertanggung jawab untuk menyelesaikan lapisan tersembunyi fungsional dalam model pembelajaran. Pilihan fungsi aktivasi ini umumnya dipilih oleh peneliti karena kinerjanya yang lebih baik, dan seringkali diterapkan pada setiap lapisan tersembunyi dari jaringan saraf. Fungsi aktivasi ReLU ini mengubah nilai input yang bersifat negatif menjadi nilai output 0, sementara untuk nilai input yang bersifat positif, outputnya sesuai dengan nilai input aktivasi tersebut [28].

2.2.6 *Object Detection*

Deteksi objek merupakan metode untuk menentukan kelas suatu objek dan memperkirakan lokasinya dengan menampilkan kotak pembatas di sekitar objek. Proses deteksi objek ini berfokus pada mengidentifikasi keberadaan objek dalam rekaman video atau gambar. Untuk melakukan deteksi dan pengukuran objek, digunakan algoritma deteksi melalui OpenCV. Pengukuran tinggi dan lebar objek dapat dihitung dengan memanfaatkan titik referensi yang dibuat oleh grid kotak pembatas [29].



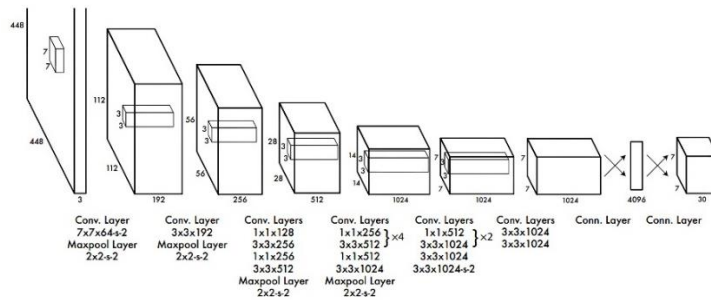
Gambar 2.7 Object Detection

Pada Gambar 2.7 menunjukkan hasil dari sistem deteksi objek pada sebuah pemandangan kota. Deteksi objek ini menggunakan *bounding boxes* berwarna untuk menandai dan mengklasifikasikan objek-objek yang teridentifikasi. Deteksi objek ini menunjukkan kemampuan model untuk mengidentifikasi dan memberi label berbagai objek dalam gambar kompleks dengan beberapa jenis objek seperti orang, mobil, tas, dan lampu lalu lintas. Model ini sangat mungkin menggunakan algoritma seperti YOLO untuk deteksi objek real-time yang cepat dan akurat.

2.2.7 You Only Look Once (YOLO)

Joseph Redmon menciptakan jaringan pendeteksi objek pada tahun 2016 yang disebut *YOLO*. Operasi YOLO relatif sederhana, menggunakan *CNN* untuk memprediksi beberapa kotak pembatas dan probabilitas kelas untuk setiap kotak. YOLO mengambil gambar input yang dibagi menjadi grid $S \times S$, yang kemudian diolah oleh jaringan saraf untuk menghasilkan kotak pembatas dan prediksi kelas. Setiap sel grid memprediksi B kotak pembatas dan skor kepercayaan untuk setiap kotak. Skor kepercayaan mencerminkan tingkat keyakinan model bahwa objek di dalam kotak sesuai dengan objek yang diprediksi. YOLO mengevaluasi kepercayaan sebagai $Pr(Object) * Intersection\ of\ Union\ (IOU_{truth})$. Setiap kotak pembatas terdiri dari 5 prediksi: x, y, w, h . Koordinat (x, y) mewakili pusat kotak relatif terhadap batas sel grid. Lebar (w) dan tinggi (h) adalah prediksi relatif untuk seluruh gambar. Setiap sel grid memprediksi probabilitas kelas bersyarat (C),

$Pr(Classi/Object)$, yang merupakan probabilitas terhubung dengan sel grid yang berisi objek [30].



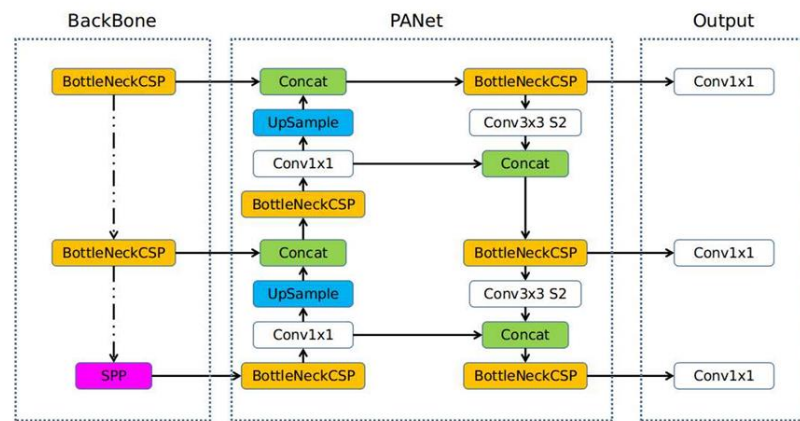
Gambar 2.8 Arsitektur *You Only Look Once (YOLO)* [31]

Pada Gambar 2.8 merupakan arsitektur *YOLO*. Sistem menerima input berupa citra berukuran $448 \times 448 \text{ pixel}$ dengan 3 channel. Citra ini kemudian mengalami satu kali proses convolutional network, menghasilkan output berukuran $(7,7,30)$. Dimana, 7×7 merupakan ukuran grid sel ($S=7$), dan 30 mencakup jumlah kotak pembatas (B) yang dikalikan dengan penjumlahan antara jumlah kelas (C) dan jumlah komponen dalam satu kotak ($B \times 5 + C$, dengan $B=2$ dan $C=20$). Selain ukuran kernel filter dan jumlah filter, dua parameter lainnya, yaitu padding dan stride yang mempengaruhi bentuk output konvolusi di seluruh prosedur.

Untuk mengurangi kehilangan data di sekitar batas gambar input, parameter padding dapat digunakan untuk menetapkan jumlah penambahan batas di sekitar tepi gambar. Hal ini penting karena, kecuali dari kernel 1×1 , sebagian besar kernel filter dalam proses konvolusi mengabaikan tepi gambar. Menerapkan *zero-padding*, yang menetapkan nilai 0 untuk setiap tepi gambar input, adalah cara yang umum dan mudah untuk mengatasi masalah. Sebaliknya, stride adalah parameter yang, ketika digunakan untuk mengecilkan output, menentukan berapa banyak pergeseran (langkah) yang akan dilakukan oleh kernel filter [31].

2.2.8 YOLOv5

YOLOv5 terdiri dari beberapa varian model, termasuk YOLOv5s, YOLOv5m, YOLOv5l, dan YOLOv5x. Huruf kecil yang mengikuti nama model menunjukkan ukuran model tersebut, yakni *small* (kecil), *medium* (sedang), *large* (besar), dan *extra-large* (sangat besar). Penggunaan model YOLOv5 yang lebih besar memerlukan spesifikasi komputer yang lebih tinggi dan berkas program yang lebih besar. Namun, model yang dihasilkan dari pelatihan akan menjadi lebih cepat dan akurat. Algoritma YOLOv5 memiliki tiga komponen utama dalam arsitekturnya, yaitu *Backbone*, *Head*, dan *Detection*.



Gambar 2. 9 Arsitektur YOLOv5 [32]

Pada Gambar 2.9, merupakan struktur arsitektur algoritma YOLOv5. *Backbone* adalah CNN yang bertugas mengumpulkan dan membentuk fitur gambar pada berbagai tingkatan detail. *Head* terdiri dari beberapa lapisan yang menggabungkan fitur gambar untuk prediksi. Sementara itu, *Detection* menggunakan fitur dari *Head* untuk melakukan prediksi kotak dan kelas [32].

2.2.9 Ikan Bawal Putih (*Pampus Argenteus*)

Bawal putih, atau dalam nama ilmiahnya *Pampus argenteus*, adalah jenis ikan laut yang termasuk dalam keluarga Stromateidae. Ikan ini biasanya ditemukan di perairan pantai dangkal, terutama di area yang berlumpur atau berpasir. Ikan bawal putih tersebar luas di Samudra Hindia dan bagian barat Samudra Pasifik, termasuk perairan di sekitar India, Indonesia, Malaysia, dan

Filipina. sebagai ikan omnivora, bawal putih memakan plankton, invertebrata kecil, dan alga. Mereka bertelur di perairan dangkal, dengan musim pemijahan yang biasanya terjadi pada musim panas. Ikan bawal putih memiliki nilai ekonomi yang tinggi karena permintaan pasar yang besar untuk dagingnya yang kaya akan protein, asam lemak omega-3, vitamin, dan mineral.



Gambar 2. 10 Ikan Bawal Putih (*Pampus Argenteus*)

Pada Gambar 2.10, dapat dilihat bahwa ikan bawal putih memiliki bentuk tubuh yang sangat pipih secara lateral (dengan punggung yang melengkung), moncong yang sangat pendek, serta sirip dada yang tidak tajam seperti pada bawal hitam dan tanpa adanya sirip perut. Sirip ekornya bercabang ke dalam, dengan bagian bawah yang lebih panjang. Warna bagian atas tubuhnya berwarna abu-abu, sementara bagian bawahnya berwarna putih dan keperakan. Sebagian besar tubuhnya memiliki bintik-bintik hitam yang sangat kecil, dan ujung sirip punggungnya memiliki warna gelap [33].

Kualitas ikan bawal putih yang bagus memiliki ciri-ciri bentuk tubuh yang sempurna (tidak ada luka). Sobekan pada mulut, luka pada sirip dan ekor, mata merah, dan ekor tidak bercabang merupakan indikator kurangnya kualitas ikan bawal putih. Kerusakan pada fisik ikan bawal putih ini bisa menyebabkan turunnya mutu dan nilai jualnya, namun ikan bawal putih yang mengalami kerusakan fisik masih sangat layak dikonsumsi. Faktor yang mempengaruhi kerusakan fisik ikan bawal putih adalah metode alat tangkap yang tidak benar, terbanting atau tergesek disarana transportasi selama proses pengangkutan dan yang sering terjadi ialah faktor cacat dari lahir [4].

2.2.10 Confusion Matrix

Confusion matrix merupakan sebuah tabel untuk mengukur permasalahan klasifikasi pada machine learning yang menampilkan jumlah prediksi benar dan salah. *Confusion matrix* dapat digunakan untuk mencari *recall*, *precision*, dan akurasi. Pada *confusion matrix* ada 4 skema yang ditampilkan, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 11 *Confusion Matrix*

Pada Gambar 2.11 terdapat penjelasan bahwa TP menggambarkan situasi di mana prediksi positif sesuai dengan kebenaran sebenarnya. Sebagai contoh, dalam kasus deteksi candi, TP berarti tidak ada ikan yang terdeteksi dan memang tidak ada ikan. FP menggambarkan situasi di mana prediksi positif tidak sesuai dengan kebenaran. Misalnya, dalam kasus sebelumnya, FP terjadi ketika terdeteksi adanya ikan padahal kenyataannya tidak ada ikan. FN menggambarkan situasi di mana prediksi negatif meskipun kebenarannya positif, yang berarti prediksi menunjukkan tidak ada ikan padahal sebenarnya ada ikan. TN menggambarkan situasi di mana prediksi negatif dan kebenarannya juga negatif, yang berarti sistem memprediksi tidak ada ikan dan memang tidak ada ikan.

2.2.11 Recall

Recall adalah evaluasi dari seberapa efektif model dalam menangkap jumlah positif aktual dengan benar mengidentifikasi mereka sebagai *true positive*. Dalam konteks yang sama, dipahami bahwa *Recall* menjadi kriteria penting dalam memilih model terbaik saat terdapat tingkat *false negative* yang signifikan. Dengan mengacu pada *confusion matrix*, *recall* dapat diketahui nilainya dengan formula sebagai berikut :

$$R = \frac{TP}{TP + FN} \dots\dots\dots(2.1)$$

Keterangan :

TP = *True Positive*

FN = *False Negative*

2.2.12 Precision

Precision membahas seberapa tepatnya model dalam memprediksi hasil positif, yaitu berapa banyak dari prediksi tersebut yang benar-benar *true positive*. *Precision* merupakan indikator yang bermanfaat terutama ketika terdapat tingkat *false positive* yang tinggi. Dengan mengacu pada *confusion matrix*, *precision* dapat diketahui nilainya dengan formula sebagai berikut:

$$P = \frac{TP}{TP + FN} \dots\dots\dots(2.2)$$

Keterangan :

TP = *True Positive*

FN = *False Negative*

2.2.13 Akurasi

Akurasi menggambarkan proporsi prediksi yang benar dari seluruh kelas yang telah diprediksi, dibandingkan dengan jumlah total data yang telah diproses. Dalam kerangka *confusion matrix*, nilai akurasi dapat dihitung menggunakan rumus berikut ini :

$$A = \frac{TP}{TP + FP + FN} \dots\dots\dots(2.3)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

FN = *False Negative*

2.2.14 Mean Average Precision (mAP)

Mean Average Precision (mAP) adalah metrik yang sering digunakan untuk mengevaluasi kinerja model deteksi objek. Metrik ini menggabungkan aspek presisi dan recall ke dalam satu nilai, memberikan gambaran umum tentang seberapa baik model dapat mendeteksi dan mengklasifikasikan objek dalam gambar. Nilai mAP yang lebih tinggi menunjukkan bahwa model memiliki kinerja yang baik dalam mendeteksi dan mengklasifikasikan objek dengan benar. Dihitung dengan mengambil rata-rata AP di semua kelas atau ambang batas IoU keseluruhan. *mAP* dapat diketahui dari formula sebagai berikut :

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \dots\dots\dots(2.4)$$

Keterangan:

AP_i = AP untuk kelas ke-i.

N = Jumlah total kelas yang dievaluasi.