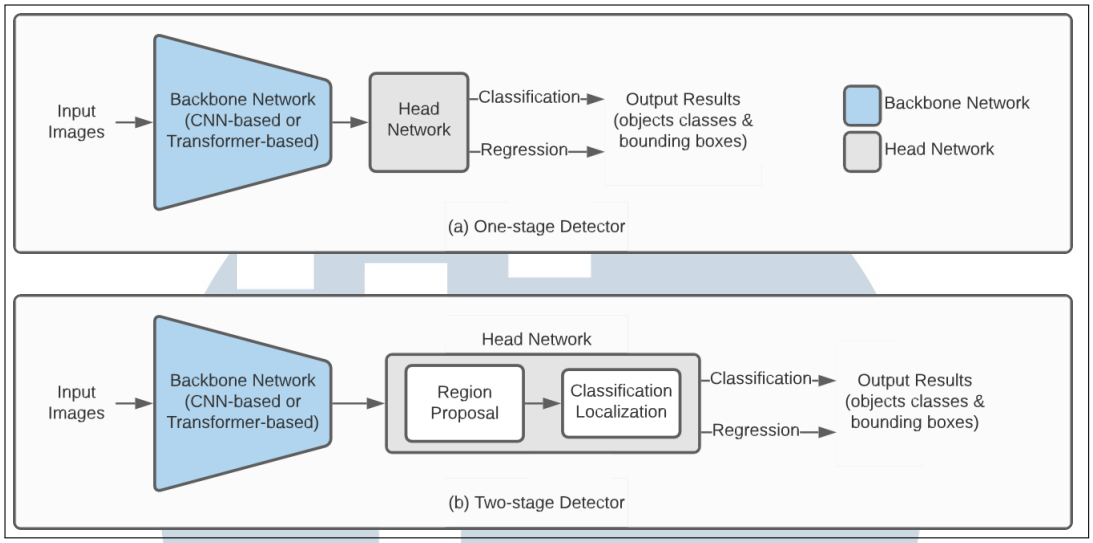
BAB 2
LANDASAN TEORI

2.1 Deteksi Objek

Deteksi objek merupakan salah satu cabang utama dalam bidang *computer vision* yang bertujuan untuk mengidentifikasi keberadaan dan lokasi suatu objek tertentu di dalam gambar atau video [38, 39]. Tidak seperti klasifikasi gambar yang hanya memprediksi kelas dari seluruh gambar, deteksi objek menghasilkan keluaran berupa *bounding box* beserta label kelas dari setiap objek yang terdeteksi [40]. Informasi lokasi ini dinyatakan dalam koordinat sudut kiri atas dan kanan bawah, atau pasangan koordinat tengah disertai lebar dan tinggi *bounding box*.

Perkembangan metode deteksi objek secara umum dapat dibagi ke dalam dua paradigma besar, yaitu metode berbasis dua tahap (*two-stage*) dan metode berbasis satu tahap (*one-stage*). Metode dua tahap seperti Faster R-CNN menghasilkan kandidat wilayah *region proposal* terlebih dahulu sebelum melakukan klasifikasi, sehingga cenderung memiliki akurasi tinggi namun kecepatan inferensi yang relatif lambat [38, 39]. Sebaliknya, metode satu tahap seperti YOLO dan SSD melakukan prediksi lokasi dan kelas secara langsung dalam satu lintasan jaringan, menghasilkan kecepatan inferensi yang jauh lebih tinggi dengan akurasi yang kompetitif [39, 40].

Gambar 2.1 mengilustrasikan perbandingan alur kerja antara metode *one-stage* dan *two-stage* dalam deteksi objek. Terlihat bahwa metode dua tahap melewati tahap *region proposal* yang terpisah sebelum melakukan klasifikasi, sedangkan metode satu tahap memprediksi seluruh keluaran secara langsung dari gambar masukan.



Gambar 2.1. Perbandingan paradigma deteksi objek *one-stage* dan *two-stage*.
Sumber: [41]

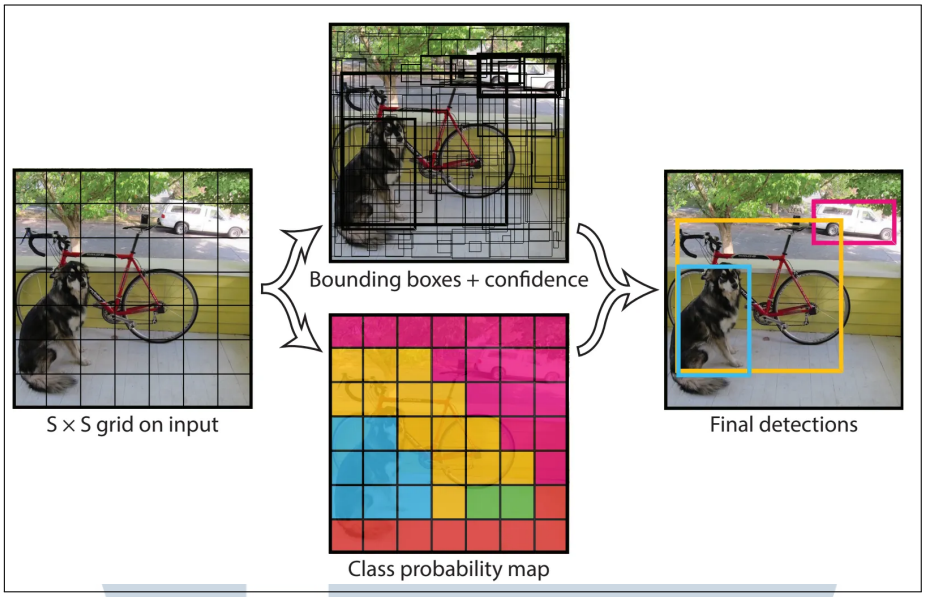
Dalam konteks aplikasi berbasis video secara waktu nyata, metode satu tahap lebih banyak dipilih karena keseimbangan antara kecepatan dan akurasi yang ditawarkannya. Kemampuan memproses gambar dalam milidetik menjadikan metode ini relevan untuk skenario pengawasan kerumunan, navigasi kendaraan otonom, dan sistem keamanan berbasis kamera [42].

2.2 You Only Look Once (YOLO)

2.2.1 Arsitektur Dasar YOLO

YOLO (*You Only Look Once*) pertama kali diperkenalkan oleh Redmon et al. [43] sebagai sebuah pendekatan revolusioner dalam deteksi objek yang memformulasikan masalah deteksi sebagai regresi tunggal. Gambar masukan dibagi menjadi grid sel berukuran $S \times S$, di mana setiap sel bertanggung jawab memprediksi sejumlah *bounding box* beserta skor kepercayaan (*confidence score*) dan distribusi probabilitas kelas. Prediksi *bounding box* mencakup koordinat pusat (x, y) yang dinormalisasi terhadap sel *grid*, serta lebar w dan tinggi h yang dinormalisasi terhadap dimensi gambar penuh.

Gambar 2.2 menggambarkan mekanisme pembagian gambar masukan menjadi *grid* pada YOLO dan cara setiap sel bertanggung jawab terhadap prediksi *bounding box* serta label kelas objek yang titik pusatnya berada di dalam sel tersebut.



Gambar 2.2. Mekanisme prediksi berbasis *grid* pada YOLO.
Sumber: [43]

Proses deteksi dalam YOLO dilakukan dalam satu lintasan jaringan saraf (*single forward pass*), yang membedakannya secara fundamental dari pendekatan berbasis proposal dua tahap. Arsitektur jaringan YOLO terdiri dari tulang punggung (*backbone*) sebagai ekstraktor fitur, leher (*neck*) untuk agregasi dan fusi fitur multiskala, serta kepala (*head*) untuk prediksi akhir. Desain ini memungkinkan YOLO mencapai kecepatan inferensi yang sangat tinggi, yang menjadikannya pilihan utama dalam aplikasi deteksi waktu nyata [44].

2.2.2 Perkembangan Arsitektur YOLO

Sejak diperkenalkan pada tahun 2016, arsitektur YOLO telah mengalami serangkaian pembaruan signifikan yang meningkatkan akurasi, kecepatan, maupun kemampuan generalisasinya. YOLOv2 memperkenalkan *anchor box* dan *batch normalization* untuk meningkatkan stabilitas pelatihan. YOLOv3 mengadopsi prediksi multiskala dengan tiga resolusi deteksi yang berbeda, sehingga lebih mampu menangani objek berukuran kecil. YOLOv4 mengintegrasikan berbagai teknik augmentasi data dan optimasi arsitektur seperti CSPNet dan PANet [39].

YOLOv5 yang dikembangkan oleh Ultralytics membawa kemudahan penggunaan yang signifikan dengan antarmuka berbasis Python serta berbagai varian ukuran model. YOLOv7 memperkenalkan *Extended Efficient Layer Aggregation Network* (E-ELAN) yang meningkatkan *gradient flow* tanpa

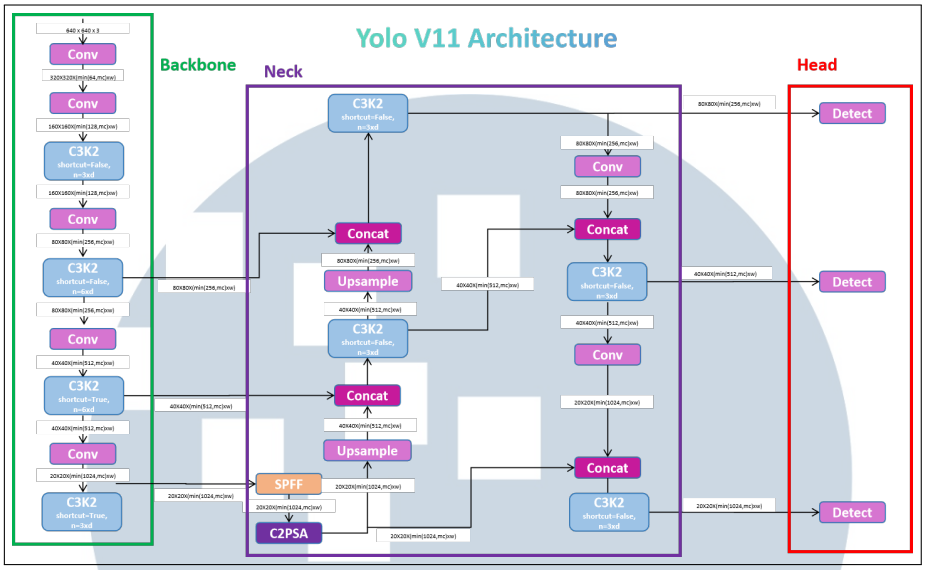
merusak pola belajar jaringan [45]. YOLOv8, yang juga dikembangkan oleh Ultralytics, mengubah kepala deteksi dari *coupled* menjadi *decoupled head* dan menghilangkan *anchor box* sehingga menjadi sepenuhnya *anchor-free* [46]. YOLOv9 memperkenalkan *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN) untuk mengatasi masalah *information bottleneck* dalam jaringan yang dalam [47].

2.2.3 Arsitektur YOLOv11

YOLOv11 merupakan salah satu generasi terbaru dari seri YOLO yang dikembangkan oleh Ultralytics dan dirilis pada tahun 2024. Arsitektur YOLOv11 mempertahankan paradigma *anchor-free* yang diperkenalkan pada YOLOv8, namun menghadirkan sejumlah pembaruan arsitektur yang meningkatkan efisiensi komputasi sekaligus akurasi deteksi [44].

Komponen utama *backbone* YOLOv11 menggunakan modul C3k2 (*Cross Stage Partial with* $k \times k$ *convolution*, *2-layered*) sebagai pengganti C2f yang digunakan pada YOLOv8. Modul C3k2 menggunakan kernel konvolusi yang lebih kecil secara berpasangan untuk mempertahankan representasi fitur yang kaya sekaligus mengurangi biaya komputasi. Selain itu, modul C2PSA (*Cross Stage Partial with Position Sensitive Attention*) diintegrasikan pada akhir *backbone* untuk meningkatkan kemampuan model dalam menangkap hubungan spasial antarobjek melalui mekanisme *self-attention* [44].

Gambar 2.3 menampilkan diagram arsitektur lengkap YOLOv11 yang mengilustrasikan susunan *backbone* dengan modul C3k2 dan C2PSA, bagian *neck* berbasis PANet, serta *decoupled detection head*.



Gambar 2.3. Arsitektur YOLOv11 dengan modul C3k2, C2PSA, dan *decoupled detection head*

Sumber: [48]

Bagian *neck* YOLOv11 menggunakan arsitektur *Path Aggregation Network* (PANet) yang diperbarui untuk fusi fitur multiskala secara *bottom-up* dan *top-down*. Fusi ini memungkinkan model untuk mendeteksi objek pada berbagai skala secara efektif, dari objek berukuran sangat kecil hingga objek berukuran besar. Kepala deteksi (*detection head*) bersifat *decoupled* dan *anchor-free*, memisahkan prediksi lokasi dan klasifikasi ke dalam dua cabang yang berbeda untuk meningkatkan akurasi masing-masing tugas [44].

YOLOv11 tersedia dalam beberapa varian ukuran, yaitu YOLOv11n (*nano*), YOLOv11s (*small*), YOLOv11m (*medium*), YOLOv11l (*large*), dan YOLOv11x (*extra-large*). Varian YOLOv11n dirancang khusus untuk skenario yang membutuhkan efisiensi komputasi tinggi dengan sumber daya terbatas, menjadikannya pilihan ideal untuk eksperimen *transfer learning* karena ukuran parameternya yang relatif kecil namun tetap kompetitif dalam hal akurasi [44].

2.3 Transfer Learning

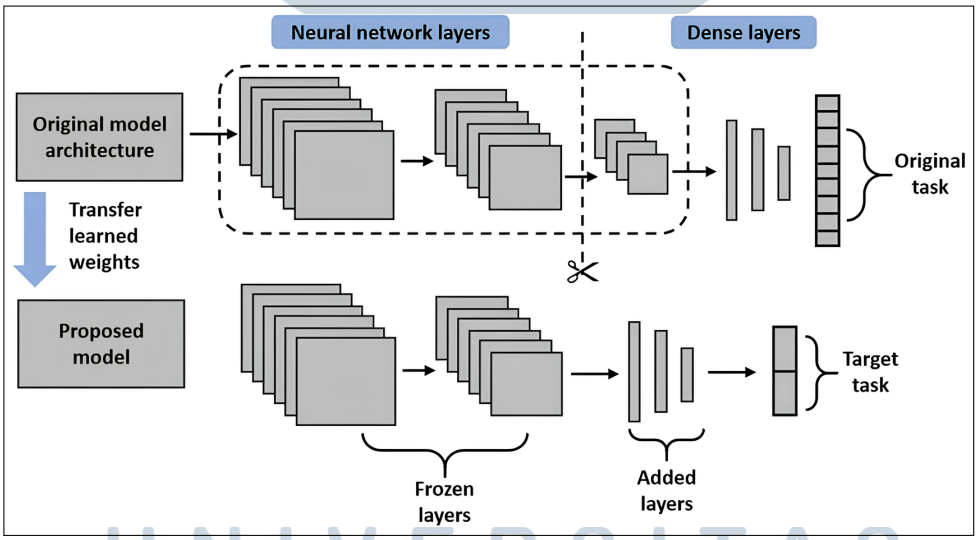
2.3.1 Konsep Dasar Transfer Learning

Transfer learning merupakan paradigma pembelajaran mesin di mana pengetahuan yang diperoleh dari pelatihan pada suatu tugas atau domain sumber (*source domain*) dimanfaatkan untuk meningkatkan kinerja model pada tugas atau

domain target (*target domain*) yang berbeda. Motivasi utama dari pendekatan ini adalah bahwa representasi fitur tingkat rendah hingga menengah, seperti detektor tepi, tekstur, dan bentuk, yang dipelajari dari dataset besar bersifat generik dan dapat digunakan lintas domain [49].

Secara formal, *transfer learning* didefinisikan dalam konteks domain $\mathscr{D}$ dan tugas $\mathscr{T}$. Domain sumber $\mathscr{D}_S$ dengan tugas $\mathscr{T}_S$ dan domain target $\mathscr{D}_T$ dengan tugas $\mathscr{T}_T$, di mana *transfer learning* bertujuan meningkatkan kinerja fungsi prediktif pada $\mathscr{T}_T$ menggunakan pengetahuan dari $\mathscr{D}_S$ dan $\mathscr{T}_S$ [49]. Dalam konteks deteksi objek berbasis *deep learning*, pengetahuan yang ditransfer umumnya berupa bobot jaringan dari model yang telah dilatih pada dataset berskala besar seperti ImageNet atau COCO.

Gambar 2.4 menunjukkan kerangka umum *transfer learning*, di mana bobot pralatih (*pretrained weights*) dari model awal yang dilatih pada tugas asal dimanfaatkan untuk menginisialisasi model pada tugas target, dengan opsi untuk membekukan (*freeze*) sebagian lapisan serta menambahkan lapisan baru guna menyesuaikan kebutuhan spesifik dari tugas target.



Gambar 2.4. Kerangka umum *transfer learning*

Sumber: [50]

Pendekatan *transfer learning* terbukti sangat bermanfaat ketika dataset target berukuran kecil atau terbatas, karena pelatihan dari awal (*training from scratch*) pada dataset kecil rentan terhadap *overfitting* dan memerlukan sumber daya komputasi yang besar. *Transfer learning* memungkinkan model untuk memulai optimisasi dari titik yang sudah bermakna secara semantik, sehingga konvergensi

lebih cepat dan generalisasi lebih baik [51].

2.3.2 Strategi Transfer Learning pada Deteksi Objek

Terdapat beberapa strategi *transfer learning* yang umum diterapkan dalam deteksi objek berbasis *deep learning*, di antaranya *feature extraction*, *fine-tuning*, dan *partial fine-tuning*.

Pada strategi *feature extraction*, seluruh lapisan *backbone* dibekukan dan hanya lapisan kepala (*head*) yang dilatih ulang menggunakan data target. Strategi ini umumnya diterapkan ketika domain sumber dan target sangat mirip, atau ketika dataset target berukuran sangat kecil [51].

Strategi *fine-tuning* menyesuaikan seluruh bobot model *pretrained* menggunakan data target, sehingga fitur tingkat tinggi dapat disesuaikan terhadap karakteristik domain target sambil mempertahankan pengetahuan fitur tingkat rendah yang bersifat generik [49]. Strategi ini umumnya menghasilkan performa optimal ketika data target memiliki ukuran yang memadai.

Strategi *partial fine-tuning* menerapkan *layer freezing*, yaitu pembekuan gradien pada sejumlah lapisan awal yang mengekstrak fitur umum seperti tepi dan tekstur, sementara lapisan- lapisan akhir yang lebih spesifik terhadap tugas dilatih ulang [51, 52]. Pendekatan ini memberikan keseimbangan antara adaptasi domain dan pencegahan *overfitting*, serta mengurangi kebutuhan komputasi pelatihan [49]. Namun, jumlah lapisan yang dibekukan merupakan hiperparameter penting: terlalu banyak pembekuan membatasi adaptasi terhadap domain target, sedangkan tanpa pembekuan sama sekali berisiko menyebabkan *catastrophic forgetting* [53]. Dalam kerangka kerja Ultralytics YOLO, parameter `freeze` digunakan untuk mengatur jumlah lapisan *backbone* yang dibekukan selama proses pelatihan [54].

Meskipun strategi *partial fine-tuning* menawarkan efisiensi komputasi dan stabilitas pelatihan, beberapa penelitian menunjukkan bahwa pendekatan ini tidak selalu menghasilkan performa optimal pada domain dengan karakteristik yang sangat berbeda. Sebagai contoh, penelitian terbaru pada model YOLO11 untuk skenario kerumunan padat dengan tingkat *occlusion* tinggi [29] menunjukkan bahwa *full fine-tuning* menghasilkan kinerja yang lebih baik dibandingkan dengan strategi *partial freezing*, khususnya dalam hal peningkatan nilai mAP dan penurunan kesalahan perhitungan objek. Temuan ini mengindikasikan bahwa pembekuan sebagian lapisan dapat membatasi kemampuan model dalam menyesuaikan representasi fitur terhadap objek berukuran kecil dan kondisi visual

yang kompleks [29].

2.3.3 Generalisasi Lintas Domain (Cross-Domain Generalization)

Generalisasi lintas domain mengacu pada kemampuan model yang telah dilatih pada suatu domain untuk mempertahankan kinerja yang baik ketika diaplikasikan pada domain lain yang memiliki distribusi data berbeda tanpa pelatihan ulang yang eksplisit. Evaluasi generalisasi lintas domain penting untuk memahami sejauh mana representasi yang dipelajari bersifat portabel dan tidak bergantung pada karakteristik spesifik suatu dataset [55].

Dalam konteks deteksi manusia pada kerumunan, uji generalisasi lintas domain dilakukan dengan melatih model pada satu dataset kerumunan (misalnya CrowdHuman) kemudian mengevaluasinya pada dataset kerumunan lain yang berbeda distribusinya (misalnya WiderPerson), dan sebaliknya. Hasil evaluasi ini memberikan informasi mengenai *robustness* model terhadap variasi kondisi pengambilan gambar, kepadatan kerumunan, dan gaya anotasi antardataset [32, 55, 56].

2.3.4 Dinamika Konvergensi dan Skala Dataset

Proses pencapaian konvergensi pada pelatihan model deteksi objek sangat dipengaruhi oleh inisialisasi bobot awal jaringan. Durasi pelatihan (jumlah *epoch*) yang lebih besar secara fundamental diperlukan oleh model yang dilatih dari awal tanpa bobot pralatih (*training from scratch*) agar representasi fitur dari ruang parameter acak dapat dipetakan menuju titik optimal [57].

Selain faktor inisialisasi, kerapatan sampel gradien per *epoch* ditentukan oleh volume direktori pelatihan. Frekuensi pembaruan gradien yang lebih sedikit dalam satu *epoch* dihasilkan oleh dataset dengan skala citra yang lebih kecil. Guna kompensasi terhadap keterbatasan iterasi per *epoch* tersebut, peningkatan alokasi *epoch* secara total mutlak diperlukan agar parameter model dapat diarahkan menuju konvergensi secara stabil tanpa terjebak pada kondisi *underfitting* [58].

2.3.5 Karakteristik Optimizer Stochastic Gradient Descent (SGD)

Proses *weight update* pada *neural network* dilakukan melalui algoritma optimasi *Stochastic Gradient Descent* (SGD). Eksekusi algoritma ini dilakukan

secara iteratif melalui subhimpunan data berukuran kecil yang disebut sebagai *mini-batch* [59]. Fluktuasi gradien dihasilkan melalui penggunaan *mini-batch* tersebut agar model dapat ditarik keluar dari jebakan *local minima* yang dangkal [59].

Mekanisme *momentum* diintegrasikan ke dalam SGD agar osilasi gradien pada *loss function* dapat diredam. Peredaman tersebut dicapai melalui penambahan sebagian nilai *update* dari iterasi sebelumnya ke dalam perhitungan iterasi saat ini secara akumulatif [59].

Penggunaan nilai koefisien *momentum* yang tinggi secara umum diaplikasikan agar stabilitas pergerakan *weight update* dapat dijaga. Pembaruan yang bersifat lebih konservatif tersebut sangat dibutuhkan pada skenario adaptasi *transfer learning* agar hilangnya *pretrained features* yang berharga dapat dicegah akibat lonjakan gradien yang terlalu agresif di awal fase pelatihan [59–61].

2.4 Deteksi Manusia pada Kerumunan Padat

2.4.1 Karakteristik Tantangan Deteksi Kerumunan

Deteksi manusia pada kondisi kerumunan padat merupakan salah satu permasalahan paling menantang dalam *computer vision*. Berbeda dengan deteksi objek pada kondisi umum, skenario kerumunan padat menghadirkan serangkaian tantangan unik yang secara fundamental membatasi kinerja detektor konvensional. Tantangan-tantangan tersebut mencakup oklusi antarpejalan kaki yang tinggi, variasi skala yang ekstrem, densitas instansi yang sangat tinggi per frame, serta deformasi penampilan yang disebabkan oleh pose tubuh yang beragam [56, 62].

Kepadatan instansi yang tinggi menyebabkan *bounding box* prediksi dari banyak individu saling tumpang tindih secara signifikan. Hal ini mengakibatkan algoritma *Non-Maximum Suppression* (NMS) konvensional cenderung menekan deteksi yang valid karena dianggap sebagai prediksi duplikat [62]. Sebagai respons terhadap permasalahan ini, berbagai modifikasi NMS telah diusulkan, seperti Soft-NMS yang mengurangi skor prediksi secara bertahap daripada menghapus prediksi secara langsung [63, 64].

2.4.2 Oklusi (Occlusion)

Oklusi merupakan kondisi di mana sebagian atau seluruh tubuh seseorang tertutupi oleh objek lain atau oleh individu lain di dalam frame gambar. Berdasarkan

sumbernya, oklusi dapat diklasifikasikan menjadi oklusi antarpejalan kaki (*inter-person occlusion*), yaitu kondisi di mana satu individu menghalangi tampilan individu lainnya, dan oklusi oleh objek latar (*background occlusion*), yaitu kondisi di mana bagian tubuh terhalang oleh elemen lingkungan seperti kendaraan atau bangunan [32].

Gambar 2.5 menampilkan contoh kondisi oklusi pada kerumunan padat, di mana terlihat bagaimana *bounding box* antar-individu saling tumpang tindih akibat jarak antar individu yang sangat rapat.

Gambar 2.5. Contoh kondisi oklusi pada gambar dari dataset CrowdHuman. Sumber: [65]

Tingkat oklusi sering dikuantifikasi menggunakan rasio *Intersection over Union* (IoU) antara *bounding box* dua individu yang berdekatan. Dataset CrowdHuman mengkategorikan instansi dengan IoU antara 0,1 hingga 0,5 sebagai oklusi sedang, dan instansi dengan IoU lebih dari 0,5 sebagai oklusi parah [56]. Tingkat oklusi yang tinggi menyebabkan fitur visual individu terpotong sehingga membuat detektor kesulitan membedakan antarbagian tubuh yang saling tumpang tindih.

Berbagai pendekatan telah dikembangkan untuk menangani oklusi dalam deteksi pejalan kaki, antara lain penerapan modul ekstraksi konteks (*context extraction module*) untuk menangkap relasi antarpejalan kaki yang saling berdekatan, modifikasi algoritma *Non-Maximum Suppression* dengan atribut densitas dan jumlah instansi yang adaptif, serta perbaikan arsitektur jaringan melalui mekanisme fusi fitur multilevel yang lebih efisien [62, 63].

2.4.3 Non-Maximum Suppression (NMS)

Non-Maximum Suppression (NMS) merupakan tahap pascapemrosesan (*post-processing*) yang esensial dalam sistem deteksi objek untuk mengeliminasi

prediksi *bounding box* yang duplikat bagi objek yang sama. Algoritma NMS bekerja dengan cara menyeleksi kandidat *bounding box* yang memiliki skor kepercayaan (*confidence score*) tertinggi, kemudian menghapus seluruh *bounding box* lain yang memiliki nilai IoU melebihi ambang batas (*threshold*) tertentu terhadap *box* terpilih tersebut [63].

Dalam skenario kerumunan padat, penerapan NMS menghadapi dilema yang fundamental. Karakteristik objek pejalan kaki yang saling berdekatan dan tumpang tindih menyebabkan nilai IoU antarindividu yang berbeda secara alami menjadi sangat tinggi. Penggunaan ambang batas IoU yang terlalu rendah (ketat) pada NMS berisiko menghapus deteksi valid pejalan kaki yang berada di latar belakang atau saling berdempetan, sehingga meningkatkan nilai *false negative*. Sebaliknya, ambang batas IoU yang terlalu tinggi dapat mempertahankan terlalu banyak *bounding box* redundan untuk satu individu yang sama [64]. Pemahaman terhadap batas toleransi NMS ini sangat krusial dalam menganalisis trade-off antara nilai *Precision* dan *Recall* pada hasil evaluasi model.

2.4.4 Teknik Augmentasi Data pada Karakteristik Kerumunan

Peningkatan ketahanan spasial model deteksi objek pada lingkungan dengan rintangan visual tinggi dapat dicapai secara artifisial melalui pipa augmentasi data. Invariansi model terhadap fluktuasi pencahayaan luar ruangan secara langsung ditingkatkan melalui manipulasi ruang warna menggunakan perturbasi *Hue, Saturation, Value* (HSV) [45].

Guna penanganan terhadap variasi skala objek, instans target diproyeksikan pada berbagai dimensi piksel menggunakan teknik augmentasi geometri seperti *translate* dan *scale*. Lebih lanjut, objek spasial berukuran kecil dalam satu *batch* secara signifikan diperbanyak melalui *mosaic augmentation* dengan penggabungan empat citra pelatihan acak menjadi satu kesatuan [66]. Kemudahan model dalam mempelajari batas-batas fitur (*decision boundaries*) dari instans manusia yang saling tumpang tindih secara ekstrem dapat ditingkatkan oleh kombinasi teknik ini dengan regularisasi *mixup* [45, 54, 66].

2.5 Dataset Deteksi Manusia

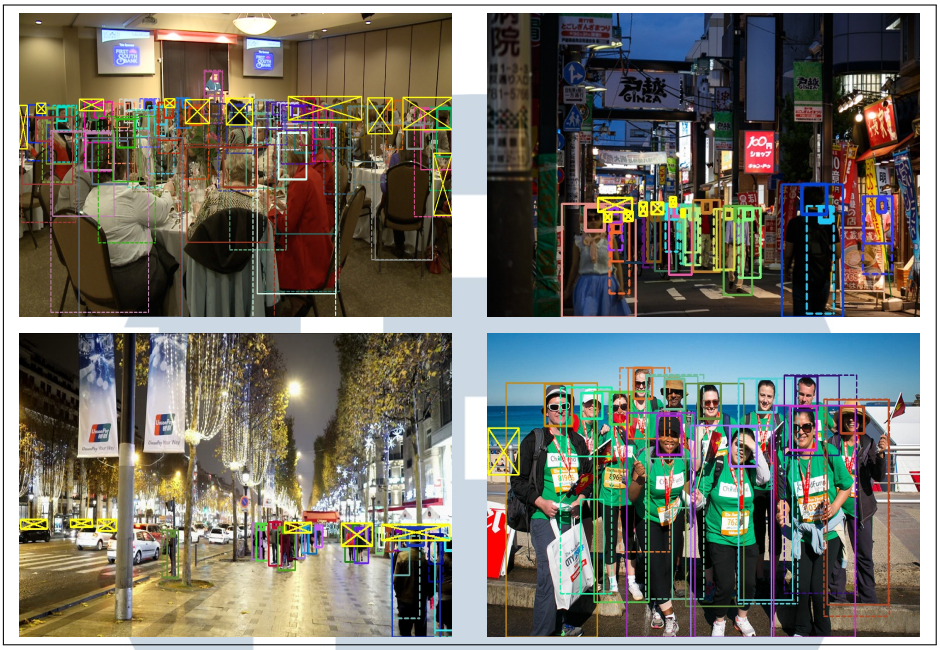
2.5.1 COCO Dataset

COCO (*Common Objects in Context*) merupakan salah satu dataset *benchmark* paling berpengaruh dalam *computer vision* yang mencakup lebih dari 330.000 gambar dengan anotasi untuk 80 kelas objek, termasuk kelas *person* [20, 67]. Dataset ini menampilkan objek dalam konteks kehidupan sehari-hari dengan variasi skala, pencahayaan, dan kondisi pengambilan gambar yang beragam. Kelas *person* dalam COCO 2017 mencakup sekitar 649.000 objek teranotasi untuk tugas deteksi objek [68].

COCO *pretrained weights* telah menjadi titik awal (*starting point*) yang paling umum digunakan dalam *transfer learning* untuk tugas deteksi objek, mengingat skala dan keragaman datanya yang sangat besar [49, 51]. Model yang dilatih pada COCO telah mempelajari representasi fitur yang kaya dan generik, mencakup berbagai tekstur, bentuk, dan konteks yang beragam. Namun demikian, distribusi kelas *person* dalam COCO tidak dirancang secara khusus untuk kondisi kerumunan padat, sehingga model yang hanya dilatih pada COCO dapat mengalami penurunan kinerja signifikan pada skenario *high occlusion* [32, 62].

2.5.2 CrowdHuman Dataset

CrowdHuman merupakan dataset *benchmark* yang dirancang secara khusus untuk menangani tantangan deteksi manusia dalam kondisi kerumunan padat dengan tingkat oklusi yang tinggi. Dataset yang diperkenalkan oleh Shao et al. [56] ini terdiri dari 15.000 gambar untuk pelatihan, 4.370 gambar untuk validasi, dan 5.000 gambar untuk pengujian, dengan total lebih dari 470.000 instansi manusia yang dianotasi.



Gambar 2.6. Contoh gambar dari dataset CrowdHuman yang menunjukkan tiga jenis *bounding box*.

Sumber: [65]

Gambar 2.6 menampilkan contoh gambar dari dataset CrowdHuman beserta anotasi tiga jenis *bounding box*, yaitu *full body* (seluruh tubuh termasuk bagian yang terhalang), *visible body* (hanya bagian tubuh yang terlihat), dan *head* (kepala). Skema anotasi yang komprehensif ini menjadi keunggulan sekaligus keunikan utama CrowdHuman dibandingkan dataset pedestrian lainnya [56].

Karakteristik CrowdHuman ditandai dengan tingkat oklusi antarpejalan kaki yang sangat tinggi, dengan rata-rata 22,6 individu per gambar dan nilai *Intersection over Union* (IoU) antarpasangan individu sebesar 0,51. Dataset ini dikumpulkan dari berbagai skenario dunia nyata seperti stasiun kereta api, tempat perbelanjaan, area konser, dan ruang publik lainnya, sehingga sangat representatif untuk melatih dan mengevaluasi model deteksi manusia [56].

2.5.3 WiderPerson Dataset

WiderPerson merupakan dataset *benchmark* untuk deteksi pejalan kaki yang dikembangkan oleh Zhang et al. [32] dengan fokus pada keragaman skenario dan kondisi yang jauh lebih luas dibandingkan dataset pejalan kaki sebelumnya. Dataset ini terdiri dari 8.000 gambar pelatihan, 1.000 gambar validasi, dan 4.382 gambar pengujian dengan total sekitar 399.786 instansi teranotasi [32].

Gambar 2.7. Contoh gambar dari dataset WiderPerson yang menunjukkan variasi tingkat kepadatan (*density*) dan keragaman skenario (*diversity*).

Sumber: [69]

Gambar 2.7 menampilkan contoh gambar dari dataset WiderPerson yang menunjukkan keragaman skenario berdasarkan lima kategori anotasi utamanya. Kategori tersebut meliputi *pedestrian*, *rider*, *partially-visible person*, *crowd*, dan *ignore region*. Pembagian ini memberikan keunggulan dalam merepresentasikan kondisi pejalan kaki dengan tingkat visibilitas dan *occlusion* yang bervariasi secara lebih akurat dalam skenario dunia nyata [32].

Karakteristik pembeda utama WiderPerson terletak pada rentang konteks pengambilan gambar yang sangat luas, mencakup skenario dalam maupun luar ruangan, variasi kondisi cuaca dan pencahayaan, serta fluktuasi kepadatan kerumunan dari rendah hingga sangat tinggi. Dengan rata-rata mencapai 29,87 anotasi per gambar, WiderPerson menjadi salah satu dataset dengan tingkat densitas anotasi tertinggi untuk evaluasi model deteksi manusia [32].

2.6 Metrik Evaluasi Deteksi Objek

2.6.1 Intersection over Union (IoU)

Intersection over Union (IoU) merupakan metrik fundamental yang digunakan dalam evaluasi kualitas prediksi *bounding box* relatif terhadap *ground truth*. IoU didefinisikan sebagai rasio antara luas irisan (*intersection*) dan luas gabungan (*union*) antara *bounding box* prediksi dan *bounding box ground truth*. Nilai IoU berkisar antara 0 dan 1, di mana kesesuaian sempurna ditunjukkan oleh nilai 1 dan tidak adanya tumpang tindih sama sekali ditandai oleh nilai 0 [70].

Secara matematis, IoU dirumuskan sebagai berikut.

$$\text{IoU} = \frac{|\mathbf{B}_{\text{pred}} \cap \mathbf{B}_{\text{gt}}|}{|\mathbf{B}_{\text{pred}} \cup \mathbf{B}_{\text{gt}}|} \tag{2.1}$$

di mana $\mathbf{B}_{\text{pred}}$ dinyatakan sebagai *bounding box* prediksi dan $\mathbf{B}_{\text{gt}}$ didefinisikan sebagai *bounding box ground truth*. Ambang batas IoU yang umum digunakan dalam penentuan kategori *True Positive* (TP) adalah IoU $\geq$ 0,5, meskipun ambang batas yang lebih ketat seperti IoU $\geq$ 0,75 juga diterapkan untuk evaluasi dengan tingkat presisi yang lebih tinggi [40].

2.6.2 Confusion Matrix

Confusion matrix merupakan tabel yang digunakan untuk menyajikan ringkasan kinerja model, di mana hasil prediksi dibandingkan terhadap *ground truth* ke dalam empat kategori utama [71]. Dalam konteks deteksi objek, pengkategorian ini didasarkan pada nilai *Intersection over Union* (IoU) antara *bounding box* prediksi dan *ground truth* terhadap ambang batas (*threshold*) yang telah ditentukan [40].

Tabel 2.1. Confusion Matrix

		Predicted Class	
		P	N
Actual Class	P	True Positive	False Negative
	N	False Positive	True Negative

Pengelompokan hasil deteksi objek berdasarkan *confusion matrix* dijabarkan sebagai berikut [40, 71].

1. *True Positive* (TP): Prediksi *bounding box* dengan nilai IoU $\geq$ ambang batas terhadap suatu *ground truth* dan berlabel kelas benar.
2. *False Positive* (FP): Prediksi *bounding box* yang dinyatakan tidak memenuhi ambang batas IoU terhadap *ground truth* mana pun, atau diklasifikasikan dengan label kelas yang salah (*false alarm*).
3. *False Negative* (FN): Instansi *ground truth* yang gagal dideteksi oleh model karena tidak ditemukannya prediksi yang berada di atas ambang batas IoU (*missed detection*).

4. *True Negative* (TN): Wilayah latar belakang (*background*) tanpa keberadaan objek dan tidak diprediksi oleh model. Dalam deteksi objek, TN tidak diperhitungkan secara eksplisit karena jumlah wilayah kosong pada gambar tidak terbatas.

2.6.3 Precision dan Recall

Precision dan *recall* merupakan dua metrik dasar yang digunakan dalam evaluasi akurasi dan kelengkapan hasil detektor pada ambang batas kepercayaan tertentu [71].

Proporsi prediksi positif model yang benar-benar akurat terhadap data asli ditunjukkan oleh metrik *precision*, yang dihitung berdasarkan Persamaan 2.2 [71].

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Sementara itu, kemampuan model dalam menemukan kembali seluruh objek target pada gambar diukur melalui metrik *recall*, yang dirumuskan pada Persamaan 2.3 [40].

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Hubungan yang saling berlawanan (*trade-off*) dicatat terjadi antara kedua metrik ini. Melalui peningkatan ambang batas skor kepercayaan model, prediksi akan disaring secara lebih ketat sehingga nilai *precision* ditingkatkan. Namun di sisi lain, nilai *recall* akan diturunkan akibat lebih banyaknya objek valid yang terlewat, dan kondisi sebaliknya didapati berlaku secara konsisten [40].

Dalam konteks deteksi kerumunan padat, *recall* dinilai sebagai metrik yang sangat krusial karena kegagalan dalam deteksi manusia (*false negative*) dapat menimbulkan dampak fatal pada sistem pengawasan keamanan publik. Namun, nilai *precision* juga harus dijaga agar tetap optimal agar produksi deteksi palsu (*false alarm*) yang berlebih dapat dihindari sehingga tidak membebani proses analisis data selanjutnya [63].

2.6.4 Average Precision (AP) dan Mean Average Precision (mAP)

Average Precision (AP) merupakan metrik agregat yang didasarkan pada luas area di bawah kurva *Precision-Recall*, di mana ringkasan performa detektor secara menyeluruh disajikan di berbagai tingkat ambang batas kepercayaan.

$$AP = \int_0^1 P(R)\, dR \tag{2.4}$$

Pada Persamaan 2.4, $P(R)$ dinyatakan sebagai nilai *precision* pada tingkat *recall* R. Performa model yang optimal ditandai oleh nilai AP yang tinggi, di mana akurasi (*precision*) yang stabil dapat dipertahankan meskipun cakupan deteksinya (*recall*) ditingkatkan [40]. Beberapa varian AP didefinisikan oleh standar evaluasi COCO berdasarkan ambang batas IoU yang digunakan, seperti AP_{50} (dengan ambang batas IoU tetap sebesar 0,5) serta $AP@[.5{:}.95]$ yang diperoleh dari perataan nilai AP pada rentang ambang batas IoU 0,5 hingga 0,95 dengan interval kenaikan 0,05 [72].

Selanjutnya, *Mean Average Precision* (mAP) digunakan dalam perhitungan rata-rata nilai AP di seluruh kelas objek yang tersedia pada sistem, sesuai dengan Persamaan 2.5.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{2.5}$$

Pada Persamaan 2.5, jumlah kategori kelas objek dinyatakan oleh N, sedangkan nilai *Average Precision* untuk kelas ke-i disimbolkan oleh AP_i. Pada penelitian dengan dataset kelas tunggal (*single-class*) seperti deteksi manusia ini, nilai mAP secara matematis bersifat ekuivalen dengan nilai AP karena proses perataan hanya dilakukan pada satu kelas objek. Meskipun demikian, istilah mAP tetap diadopsi dalam pelaporan hasil guna menjaga konsistensi standarisasi metrik pada tugas deteksi objek.