LAPORAN MINI PROJECT
CSCE604133 Computer Vision

Pendeteksi Mario dengan Pendekatan *Object Detection* Menggunakan Model YOLO11

2206830542 - Muhammad Hilal Darul Fauzan
2206028251 - Patrick Samuel Evans Simanjuntak

Semester Ganjil 2024/2025
Fakultas Ilmu Komputer
Universitas Indonesia

Contents

Chapter 1

Pendahuluan

Teknologi *object detection* merupakan salah satu cabang dari *computer vision* yang memiliki peran penting dalam berbagai aplikasi, seperti pengenalan wajah, sistem pengawasan, kendaraan otonom, dan pengolahan video berbasis kecerdasan buatan. Dengan kemampuan untuk mengenali dan melokalisasi objek tertentu dalam gambar atau video, *object detection* telah menjadi topik penelitian yang berkembang pesat, terutama dengan kemajuan teknologi deep learning.

Pada proyek ini, implementasi *object detection* difokuskan pada pendeteksian karakter Mario dari video yang telah disediakan pada dataset. Pendeteksian ini bertujuan untuk mengevaluasi bagaimana algoritma YOLO (*You Only Look Once*) dapat digunakan untuk mendeteksi objek tertentu dengan akurasi yang tinggi dan waktu pemrosesan yang efisien. YOLO sebagai salah satu algoritma *object detection* yang populer, memiliki keunggulan dalam deteksi *real time* karena kemampuannya untuk memproses gambar dalam satu langkah komputasi tanpa memerlukan tahap ekstraksi fitur terpisah.

1.1 Deskripsi Masalah

Kebutuhan untuk menganalisis data visual, seperti gambar dan video telah meningkat secara signifikan. Proyek ini dilatarbelakangi oleh tantangan utama dalam pengolahan data visual adalah mendeteksi dan mengenali objek tertentu dalam suatu frame video dengan akurasi yang tinggi dan waktu pemrosesan yang efisien. Pendeteksian objek ini tidak hanya memerlukan algoritma yang canggih, tetapi juga memerlukan optimasi untuk memastikan bahwa sistem dapat berfungsi dengan baik dalam skenario dunia nyata yang penuh dengan variabilitas, seperti perbedaan pencahayaan, sudut pandang, dan kompleksitas latar belakang.

Proyek ini dimulai dari permasalahan tersebut, dengan fokus pada penerapan *object detection* untuk mendeteksi objek spesifik, yaitu karakter Mario pada video. Tantangan utamanya melibatkan pengembangan sistem yang mampu mendeteksi objek target secara akurat sekaligus mempertahankan kecepatan

pemrosesan.

1.2 Tujuan

Proyek ini bertujuan untuk mengembangkan dan mengimplementasikan sistem *object detection* berbasis YOLO (*You Only Look Once*) yang mampu mendeteksi objek Mario pada dataset video yang diberikan. Sistem ini dirancang agar mampu memberikan hasil deteksi yang akurat dengan membandingkan hasil pendeteksian terhadap *ground truth* yang telah ditentukan. Evaluasi sistem akan dilakukan menggunakan dua metrik utama, yaitu akurasi dan kecepatan/kompleksitas. Akurasi diukur untuk menilai ketepatan model dalam mengenali dan melokalisasi objek target, sedangkan kecepatan dan kompleksitas diukur untuk menilai efisiensi pemrosesan sistem.

Proyek ini juga bertujuan untuk memberikan pengalaman secara langsung dalam mempelajari, mengimplementasikan, dan mengevaluasi algoritma *object detection*. Dengan pendekatan ini diharapkan dapat memahami bagaimana metode YOLO bekerja dalam konteks data video.

1.3 Batasan Penelitian

Penelitian ini memiliki beberapa batasan yang ditetapkan untuk memastikan fokus dan cakupan yang jelas dalam pengembangan sistem *object detection*. Objek yang menjadi target deteksi adalah karakter Mario, yang berarti penelitian ini terbatas pada skenario pendeteksian objek tersebut dalam video. Data video yang digunakan akan disediakan sehingga sistem tidak dirancang untuk menangani data video di luar dataset yang disediakan atau skenario yang tidak terduga. Algoritma yang digunakan untuk implementasi *object detection* adalah YOLO (*You Only Look Once*). Dengan demikian, penelitian ini tidak mencakup eksplorasi atau perbandingan dengan algoritma lain seperti SSD (Single Shot Detector) atau Faster R-CNN. Fokus utama adalah pada penerapan dan evaluasi performa YOLO dalam mendeteksi objek Mario.

Evaluasi performa sistem juga dibatasi pada dua kriteria utama, yaitu akurasi dan kecepatan/kompleksitas. Akurasi diukur dengan membandingkan hasil deteksi terhadap *ground truth* yang telah disediakan, sementara kecepatan dievaluasi dalam konteks waktu pemrosesan. Penelitian ini tidak mencakup aspek lain seperti konsumsi daya atau adaptabilitas sistem terhadap perangkat keras tertentu. Batasan lainnya adalah pendekatan yang digunakan dalam desain eksperimen. Penelitian ini dirancang untuk memberikan hasil evaluasi yang mencerminkan kemampuan algoritma dalam mendeteksi objek target tanpa melibatkan skenario dunia nyata yang lebih kompleks, seperti variasi pencahayaan ekstrem, latar belakang bergerak, atau resolusi video yang berbeda-beda. Oleh karena itu, hasil penelitian ini lebih relevan untuk skenario pengujian yang terkontrol dibandingkan aplikasi langsung di lingkungan produksi.

Chapter 2

Studi Literatur

2.1 Konsep *Object Detection*

Object detection adalah bagian dari *computer vision* yang menggabungkan tugas-tugas visi kompleks, seperti segmentasi, pengenalan, dan lokalisasi menjadi satu masalah terintegrasi (Hu et al., 2023). Tujuan utamanya adalah mengidentifikasi dan menemukan objek dalam gambar dengan memberikan kelas objek dan posisinya melalui *bounding boxes* (Hu et al., 2023). Hal ini sangat penting untuk berbagai aplikasi, termasuk kendaraan otonom, sistem pengawasan, dan pengambilan gambar.

Algoritma *object detection* dapat dikategorikan secara luas menjadi dua jenis utama: metode dua tahap dan satu tahap (Hu et al., 2023). Algoritma dua tahap seperti Faster R-CNN dan Mask R-CNN pertama menghasilkan proposal wilayah kemudian mengklasifikasikannya, mencapai akurasi tinggi dengan mengorbankan kecepatan. Sebaliknya, metode satu tahap seperti seri YOLO (You Only Look Once) menghilangkan tahap proposal wilayah, memungkinkan deteksi lebih cepat dengan memprediksi bounding boxes dan probabilitas kelas langsung dari seluruh gambar (Hu et al., 2023).

Seri YOLO telah menjadi salah satu kerangka object detection paling populer karena efisiensi dan kecepatannya (Hu et al., 2023). Model YOLO dirancang untuk memproses gambar dalam satu kali proses, secara signifikan mengurangi beban komputasi dibandingkan model dua tahap. Versi selanjutnya seperti YOLOv3 dan YOLOv4 telah menggabungkan perbaikan seperti jaringan backbone yang lebih dalam, feature pyramid networks (FPN), dan teknik augmentasi data canggih, meningkatkan akurasi dan ketangguhan dalam mendeteksi objek di berbagai skala dan lingkungan (Hu et al., 2023).

Meskipun terdapat kemajuan signifikan namun tetap ada tantangan, terutama dalam mendeteksi objek kecil dalam gambar resolusi tinggi, seperti citra udara (Hu et al., 2022; Yan et al., 2022). Objek kecil sering hanya menempati beberapa piksel, membuatnya sulit bagi model untuk mempelajari fitur mereka secara efektif. Algoritma object detection tradisional yang dirancang untuk pe-

mandangan alami mungkin tidak berfungsi baik dalam konteks udara, di mana objek tersebar padat dan bervariasi secara signifikan dalam ukuran (Hu et al., 2023).

Integrasi teknologi object detection dengan bidang yang berkembang seperti pemantauan satwa liar menunjukkan keserba-gunaan teknik ini. Para peneliti telah berhasil menerapkan model seperti YOLO untuk mendeteksi spesies tertentu dalam citra termal udara, memungkinkan pemantauan real-time dan berkontribusi pada upaya konservasi (Povlsen et al., 2023). Seiring bidang ini terus berkembang, penelitian berkelanjutan berfokus pada mengembangkan model yang meningkatkan deteksi objek kecil sambil menyeimbangkan akurasi dan efisiensi komputasi, memastikan relevansi berkelanjutan teknologi object detection dalam lanskap yang terus berubah (Hu et al., 2023).

2.2 Algoritma YOLO dalam *Object Detection*

YOLO adalah sistem deteksi objek real-time yang populer yang memproses gambar dalam satu kali pemrosesan, membuatnya jauh lebih cepat dibandingkan dengan metode tradisional yang memerlukan beberapa kali pemrosesan terhadap gambar (Yan et al., 2022). Arsitektur asli YOLO telah mengalami beberapa iterasi, dengan YOLOv5 menjadi salah satu versi terbaru (Yan et al., 2022). YOLOv5 menggunakan Path Aggregation Network (PANet) untuk mengurangi kehilangan informasi selama ekstraksi fitur, yang sangat penting untuk mendeteksi objek kecil yang membawa sedikit informasi (Yan et al., 2022).

You Only Look Once (YOLO) telah muncul sebagai kerangka kerja terkemuka dalam bidang deteksi objek, terutama dikenal karena efisiensinya dan kemampuan pemrosesan real-time (Hu et al., 2023). YOLO beroperasi sebagai model deteksi objek satu tahap, yang berarti model ini memprediksi baik kotak pembatas maupun probabilitas kelas secara langsung dari gambar input dalam satu kali evaluasi (Hu et al., 2023). Pendekatan ini berbeda dengan model dua tahap tradisional yang pertama-tama menghasilkan proposal wilayah dan kemudian mengklasifikasikannya, yang mengarah pada waktu pemrosesan yang lebih lambat (Hu et al., 2023). Arsitektur YOLO memungkinkannya untuk mencapai kecepatan tinggi sambil mempertahankan tingkat akurasi yang tinggi (Hu et al., 2023).

Kerangka deteksi objek YOLO (You Only Look Once) merupakan pendekatan yang menonjol dan efisien dalam bidang computer vision, khususnya untuk aplikasi real-time (Povlsen et al., 2023). Model YOLO dirancang untuk melakukan deteksi objek dan segmentasi gambar dengan memperlakukan tugas ini sebagai masalah regresi tunggal (Povlsen et al., 2023). Ini berarti bahwa alih-alih memindai gambar beberapa kali untuk mengidentifikasi objek, YOLO memproses seluruh gambar dalam satu kali proses, memprediksi kotak pembatas dan probabilitas kelas secara bersamaan (Povlsen et al., 2023). Pendekatan unik ini memungkinkan pemrosesan yang cepat, menjadikan YOLO cocok untuk aplikasi yang membutuhkan umpan balik segera, seperti pengawasan dan sistem otonom (Povlsen et al., 2023).

Dalam konteks penelitian ini menggunakan kasus pemantauan satwa liar, YOLO telah terbukti efektif saat digunakan untuk meningkatkan deteksi spesies tertentu dalam rekaman udara (Povlsen et al., 2023). Studi yang dirujuk dalam dokumen ini menyoroti penggunaan YOLOv5, salah satu versi dari kerangka YOLO, untuk mengidentifikasi titik minat (POI), serta hewan-hewan spesifik seperti kelinci dan rusa roe dalam gambar termal (Povlsen et al., 2023). Kemampuan untuk mendeteksi spesies ini secara real-time menggunakan teknologi drone dapat secara signifikan meningkatkan upaya konservasi dengan menyediakan data yang tepat waktu tentang perilaku hewan dan dinamika populasi (Povlsen et al., 2023). Integrasi YOLO dengan kendaraan udara tanpa awak (UAV) memungkinkan pemantauan yang efisien di area luas, yang sangat bermanfaat untuk mempelajari satwa liar yang sulit dijangkau atau nokturnal (Povlsen et al., 2023).

Kemampuan adaptasi model YOLO merupakan keuntungan utama lainnya karena model ini dapat dilatih pada dataset kustom yang disesuaikan dengan lingkungan atau spesies tertentu (Povlsen et al., 2023). Fleksibilitas ini memungkinkan para peneliti untuk meningkatkan akurasi model dalam mendeteksi hewan tertentu di bawah berbagai kondisi (Povlsen et al., 2023). Studi ini menekankan bahwa efektivitas model yang dilatih sangat bergantung pada kualitas dan relevansi dataset pelatihan (Povlsen et al., 2023).

Secara keseluruhan, YOLO merupakan kemajuan signifikan dalam teknologi deteksi objek, khususnya dalam bidang pemantauan satwa liar (Povlsen et al., 2023). Kemampuannya untuk memberikan analisis real-time dan adaptabilitasnya terhadap berbagai lingkungan menjadikannya alat yang berharga bagi para peneliti dan konservasionis (Povlsen et al., 2023). Integrasi YOLO dengan teknologi drone tidak hanya meningkatkan efisiensi pengumpulan data tetapi juga mendukung pengambilan keputusan yang lebih baik dalam manajemen satwa liar dan upaya konservasi (Povlsen et al., 2023).

2.3 YOLOv11 dalam *Object Detection*

YOLOv11 adalah iterasi terbaru dalam seri model deep learning YOLO (You Only Look Once) yang secara khusus dirancang untuk tugas deteksi objek (Alif, 2024). Model ini dibangun berdasarkan kemajuan yang dicapai oleh pendahulunya, yaitu YOLOv8 dan YOLOv10, dengan memperkenalkan beberapa peningkatan arsitektur yang meningkatkan kecepatan deteksi, akurasi, dan ketahanan dalam lingkungan yang kompleks (Alif, 2024). YOLOv11 sangat cocok untuk aplikasi real-time, sehingga menjadi alat yang berharga di bidang seperti pengemudian otonom, sistem lalu lintas cerdas, dan pengawasan perkotaan (Alif, 2024). YOLOv11 juga menekankan efisiensi dan skalabilitas, menawarkan berbagai ukuran model dari nano hingga extra-large (Khanam & Hussain, 2024). Skalabilitas ini memungkinkan penerapan di berbagai lingkungan, mulai dari perangkat edge dengan sumber daya terbatas hingga sistem komputasi berkinerja tinggi (Khanam & Hussain, 2024). Desain arsitektur memastikan bahwa bahkan model yang lebih kecil pun tetap mempertahankan keseimbangan yang

baik antara efisiensi komputasi dan akurasi deteksi (Khanam & Hussain, 2024). Kinerja YOLOv11 mencapai peningkatan signifikan dalam mean Average Precision (mAP) dan kecepatan inferensi dibandingkan dengan versi sebelumnya (Khanam & Hussain, 2024). Jumlah parameter yang lebih efisien pada model ini berkontribusi pada pemrosesan yang lebih cepat tanpa mengorbankan akurasi, yang penting untuk aplikasi yang memerlukan analisis real-time (Khanam & Hussain, 2024).

Dari segi kecepatan inferensi, YOLOv11 mencapai 290 frame per detik (FPS), melampaui baik YOLOv8 maupun YOLOv10 (Alif, 2024). Kecepatan inferensi yang tinggi ini, dikombinasikan dengan kemampuan deteksi yang kuat, menjadikan YOLOv11 sebagai solusi yang ampuh untuk aplikasi deteksi objek real-time (Alif, 2024). Ketahanan model ini semakin terlihat dari performanya dalam berbagai kondisi lingkungan dan ukuran objek yang berbeda (Alif, 2024). YOLOv11 menunjukkan konsistensi yang lebih tinggi dalam mengidentifikasi dan mengklasifikasikan objek yang terhalang sebagian atau tidak terlihat sepenuhnya (Alif, 2024). Dari segi fleksibilitas, YOLOv11 memperluas utilitasnya di luar tugas deteksi objek tradisional untuk mencakup instance segmentation, pose estimation, dan oriented object detection (Khanam & Hussain, 2024). Kemampuan multi-task ini menjadikan YOLOv11 sebagai solusi komprehensif untuk berbagai tantangan computer vision (Khanam & Hussain, 2024). Model ini tersedia dalam berbagai ukuran, mulai dari nano hingga extra-large, yang memenuhi kebutuhan aplikasi yang berbeda dan memungkinkan penerapan di lingkungan dengan sumber daya terbatas tanpa mengorbankan kinerja (Khanam & Hussain, 2024).

Arsitektur model ini menggabungkan lapisan-lapisan canggih dan mekanisme yang ditingkatkan memungkinkan dalam menangani tantangan dalam mendeteksi objek yang lebih kecil dan terhalang (Alif, 2024). Salah satu kekuatan utama YOLOv11 adalah kemampuannya mempertahankan presisi dan recall yang tinggi di berbagai jenis objek (Alif, 2024). Pada penelitian Alif (2024) performa YOLOv11 dievaluasi menggunakan metrik seperti precision, recall, F1 score, dan mean Average Precision (mAP). Hasil evaluasi menunjukkan bahwa YOLOv11 melampaui pendahulunya, terutama dalam mendeteksi objek yang lebih kecil dan lebih kompleks (Alif, 2024). Pada inti arsitektur YOLOv11 terdapat elemen baru seperti C3k2 block, Spatial Pyramid Pooling - Fast (SPPF), dan Convolutional block with Parallel Spatial Attention (C2PSA) (Khanam & Hussain, 2024). Komponen-komponen ini bekerja bersama untuk meningkatkan kemampuan ekstraksi dan pemrosesan fitur, memungkinkan model menganalisis informasi visual yang kompleks secara lebih efektif (Khanam & Hussain, 2024). Sebagai contoh, C3k2 block mengoptimalkan aliran informasi melalui jaringan, sementara SPPF meningkatkan kemampuan model untuk menangani berbagai ukuran input, sehingga meningkatkan ketangguhannya di berbagai skenario (Khanam & Hussain, 2024).

Selain peningkatan arsitektur, YOLOv11 menggunakan paradigma deteksi anchor-free, yang menyederhanakan proses deteksi dengan menghilangkan kebutuhan akan kotak jangkar yang telah ditentukan sebelumnya (Alif, 2024). Pendekatan ini memungkinkan model untuk beradaptasi lebih fleksibel den-

gan berbagai bentuk dan ukuran objek, sehingga meningkatkan kemampuan deteksinya lebih jauh (Alif, 2024). Kombinasi dari inovasi arsitektur ini menghasilkan model yang tidak hanya mencapai akurasi tinggi tetapi juga mempertahankan kecepatan inferensi yang kompetitif, dengan YOLOv11 mencapai hingga 290 frame per detik (FPS) (Alif, 2024). Kecepatan ini sangat penting untuk aplikasi waktu nyata, di mana pengambilan keputusan yang cepat menjadi kritikal (Alif, 2024).

Salah satu fitur unggulan YOLOv11 adalah jumlah parameter yang lebih efisien, yang memungkinkan kinerja model yang lebih cepat tanpa secara signifikan memengaruhi akurasi keseluruhan (Khanam & Hussain, 2024). Pengurangan parameter ini sangat bermanfaat untuk aplikasi real-time di mana kecepatan dan akurasi sangat penting (Khanam & Hussain, 2024). Model ini juga dilengkapi dengan kemampuan ekstraksi fitur yang canggih, berkat peningkatan pada arsitektur backbone dan neck-nya (Khanam & Hussain, 2024). Peningkatan ini memungkinkan YOLOv11 menganalisis dan menginterpretasikan informasi visual yang kompleks secara lebih efektif, yang mengarah pada hasil deteksi objek yang lebih presisi (Khanam & Hussain, 2024). Benchmark kinerja menunjukkan bahwa YOLOv11 secara konsisten mengungguli pendahulunya, mencapai skor mean Average Precision (mAP) yang lebih tinggi pada dataset seperti COCO sambil mempertahankan tingkat inferensi yang lebih cepat (Khanam & Hussain, 2024). Sebagai contoh, varian YOLOv11x mencapai sekitar 54,5% mAP dengan latensi 13ms, melampaui semua iterasi YOLO sebelumnya (Khanam & Hussain, 2024). Namun, meski memiliki banyak keunggulan, YOLOv11 juga memiliki beberapa keterbatasan (Alif, 2024). Model ini kadang kesulitan membedakan antara jenis objek yang secara visual mirip (contoh : truk dan bus) terutama ketika objek tersebut terhalang sebagian atau dilihat dari sudut yang sulit (Alif, 2024).

Secara keseluruhan, kombinasi ekstraksi fitur yang ditingkatkan, kinerja yang dioptimalkan, dan dukungan untuk berbagai tugas menjadikan YOLOv11 sebagai solusi tangguh untuk menghadapi tantangan pengenalan visual yang kompleks (Khanam & Hussain, 2024). Implikasinya untuk berbagai industri, termasuk kendaraan otonom, sistem pengawasan, dan otomasi industri, sangat signifikan, karena menawarkan efisiensi dan adaptabilitas yang lebih baik di berbagai aplikasi (Khanam & Hussain, 2024). Kemajuan pada YOLOv11 tidak hanya mendorong batas teknologi deteksi objek real-time tetapi juga membuka kemungkinan baru untuk penerapan di berbagai lingkungan, menjadikannya alat yang berharga untuk penelitian maupun aplikasi praktis (Khanam & Hussain, 2024).

Chapter 3

Metodologi

3.1 Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini berisi gambar-gambar yang menampilkan objek Mario, yang diperoleh dari berbagai sumber. Sumber utama dataset ini meliputi data pelatihan (train) yang disediakan oleh platform Scele, video open test yang disampling setiap 30 frame, gambar yang diambil melalui pencarian di Google, serta gambar-gambar yang dicetak dan difoto. Secara keseluruhan, dataset ini terdiri dari 570 gambar.

Proses anotasi dilakukan pada platform Roboflow, di mana setiap objek Mario yang terdeteksi diberi label dengan bounding box yang dikategorikan sebagai kelas "mario". Setelah proses pengumpulan dan anotasi, dataset dibagi menjadi tiga bagian dengan proporsi 8:1:1 untuk data pelatihan, validasi, dan pengujian. Secara rinci, dataset pelatihan terdiri dari 454 gambar, dataset validasi sebanyak 58 gambar, dan dataset pengujian juga sebanyak 58 gambar.

Selanjutnya, untuk meningkatkan jumlah data pelatihan, dilakukan augmentasi gambar pada dataset pelatihan melalui Roboflow, yang menghasilkan total 3178 gambar. Sebelum proses augmentasi, setiap gambar pada dataset menjalani preprocessing, yang meliputi dua tahap utama: auto-orient (penyesuaian orientasi gambar) dan resize menjadi 480x480 piksel. Proses resize dilakukan untuk mengurangi kompleksitas komputasi serta menyesuaikan ukuran objek Mario yang relatif besar dalam gambar.

Augmentasi yang diterapkan pada dataset ini dibagi menjadi dua kategori, yaitu image-level augmentations dan bounding box-level augmentations. Penjelasan rinci mengenai kedua jenis augmentasi ini adalah sebagai berikut:

3.1.1 Image-Level Augmentations

Image-level augmentations adalah teknik augmentasi yang diterapkan langsung pada gambar tanpa memperhitungkan posisi atau bentuk bounding box. Tujuan dari augmentasi ini adalah untuk meningkatkan keberagaman gambar yang digunakan dalam pelatihan, sehingga model dapat lebih robust dan mampu men-

genali objek Mario dalam berbagai variasi gambar. Beberapa teknik image-level augmentations yang diterapkan dalam penelitian ini meliputi:

- Flip horizontal dan vertikal: Membalik gambar secara horizontal atau vertikal untuk menghasilkan variasi gambar yang berbeda.
- Rotasi: Memutar gambar dengan rentang sudut antara -13° hingga +13° untuk menciptakan variasi sudut pandang.
- Shear: Melakukan pergeseran gambar secara horizontal dan vertikal dalam rentang -15° hingga +15° untuk mensimulasikan perubahan perspektif.
- Grayscale: Mengubah gambar menjadi grayscale dengan tingkat keabuan hingga 15% dari gambar asli.
- Perubahan hue: Mengubah warna gambar dengan rentang -25 hingga +25 derajat untuk memperkenalkan variasi warna.
- Perubahan saturasi: Menurunkan atau meningkatkan saturasi warna gambar antara -25% hingga +25%.
- Perubahan brightness: Mengatur tingkat kecerahan gambar antara -25% hingga +25%.
- Noise: Menambahkan noise pada gambar hingga 0,5% untuk meningkatkan robustness model terhadap gangguan visual.

3.1.2 Bounding Box-Level Augmentations

Bounding box-level augmentations adalah teknik augmentasi yang diterapkan pada gambar dengan mempertimbangkan posisi dan ukuran bounding box yang mengelilingi objek yang dianotasi. Teknik ini memastikan bahwa perubahan pada gambar tetap mempertahankan konsistensi antara objek dan labelnya. Augmentasi pada tingkat bounding box yang diterapkan dalam penelitian ini meliputi:

- Bounding box flip horizontal dan vertikal: Melakukan pembalikan posisi bounding box secara horizontal atau vertikal sesuai dengan perubahan gambar.
- Bounding box shear: Menggeser posisi bounding box secara horizontal dan vertikal dalam rentang -10° hingga +10° untuk mensimulasikan perubahan perspektif.
- Bounding box brightness: Mengubah tingkat kecerahan gambar dan menyesuaikan posisi bounding box dalam rentang -15% hingga +15%.
- Blur: Menambahkan efek blur pada gambar dengan tingkat ketajaman hingga 2,5 piksel, yang juga mempengaruhi posisi dan ukuran bounding box.

- Noise pada bounding box: Menambahkan noise kecil (hingga 0,1%) pada koordinat bounding box untuk meningkatkan robustness model terhadap variansi data.

Dengan penerapan teknik-teknik augmentasi ini, diharapkan model dapat dilatih untuk mengenali objek Mario secara lebih efektif, meskipun terdapat variasi dalam kondisi gambar atau gangguan eksternal.

3.2 Desain Eksperimen

Dataset yang telah dipersiapkan digunakan untuk melatih beberapa varian model YOLO (You Only Look Once), yaitu YOLOv11 dengan arsitektur model YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, dan YOLOv11x. YOLO merupakan model pretrained, yang artinya model ini telah dilatih sebelumnya pada dataset besar seperti COCO atau VOC, untuk mendeteksi berbagai objek umum. Proses pelatihan ini dilakukan pada platform Kaggle dengan menggunakan GPU NVIDIA P100 untuk mempercepat komputasi.

3.2.1 Konfigurasi Pelatihan

Proses pelatihan menggunakan konfigurasi hyperparameter berikut:

- **epochs**: Jumlah epoch pelatihan sebanyak 100.
- **imgsz**: Ukuran gambar yang digunakan selama pelatihan adalah 480 piksel.
- **device**: GPU pertama digunakan untuk pelatihan, yaitu dengan nilai `device=0`.
- **optimizer**: Optimizer yang digunakan adalah AdamW, dengan konfigurasi berikut:
  - Learning rate (`lr`): 0.002
  - Momentum: 0.9
  - Weight decay: 0.0005

3.2.2 Augmentasi Data

Selama pelatihan, augmentasi data dilakukan untuk meningkatkan keragaman dataset dan mencegah overfitting. Teknik augmentasi yang digunakan meliputi:

- **Blur**: Probabilitas 0.01, dengan batas blur antara 3 hingga 7 piksel.
- **MedianBlur**: Probabilitas 0.01, dengan batas blur antara 3 hingga 7 piksel.
- **ToGray**: Probabilitas 0.01, mengubah gambar menjadi grayscale.

- **CLAHE (Contrast Limited Adaptive Histogram Equalization)**: Probabilitas 0.01, dengan `clip limit` antara 1 hingga 4, dan ukuran grid 8x8.

3.2.3 Metode Evaluasi

Kinerja model dievaluasi menggunakan metrik berikut:

- **Precision**: Mengukur rasio deteksi yang benar terhadap total deteksi yang dilakukan oleh model. Rumus precision adalah:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: Mengukur rasio deteksi yang benar terhadap total objek yang seharusnya terdeteksi. Rumus recall adalah:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **mAP@50 (Mean Average Precision at IoU 0.50)**: Menghitung rata-rata *average precision (AP)* pada semua kelas dengan ambang batas Intersection over Union (IoU) sebesar 0.50. Rumus mAP@50 adalah:

$$\text{mAP@50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i(IoU \geq 0.50)$$

- **mAP@0.95 (Mean Average Precision at IoU 0.50 to 0.95)**: Menghitung rata-rata *average precision (AP)* pada semua kelas dengan IoU yang bervariasi dari 0.50 hingga 0.95 (interval 0.05). Rumus mAP@0.95 adalah:

$$\text{mAP@0.95} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i(IoU \in [0.50, 0.95])$$

Chapter 4

Hasil dan Analisis

4.1 Hasil Eksperimen

Pada eksperimen ini, berbagai varian model YOLOv11 dengan arsitektur v11n, v11s, v11m, v11l, dan v11x diuji untuk mendeteksi objek pada dataset yang telah dipersiapkan. Hasil dari pelatihan dan validasi menunjukkan bahwa secara umum, semua model memberikan kinerja yang sangat baik, dengan nilai precision dan recall yang hampir sempurna (1.0 pada recall dan 0.978–0.979 pada precision). Namun, perbedaan yang signifikan muncul pada metrik mAP50 dan mAP50-95, serta pada waktu pelatihan yang dibutuhkan untuk masing-masing model.

Table 4.1: Training Metrics

	v11n	**v11s**	**v11m**	**v11l**	**v11x**
Precision	0.978	0.979	0.979	0.979	0.978
Recall	1	1	1	1	1
mAP50	0.991	0.988	0.990	0.991	0.992
mAP50-95	0.985	0.988	0.915	0.915	0.915

Table 4.2: Validation Metrics

	v11n	**v11s**	**v11m**	**v11l**	**v11x**
Precision	0.978	0.979	0.979	0.979	0.978
Recall	1	1	1	1	1
mAP50	0.991	0.988	0.990	0.991	0.992
mAP50-95	0.904	0.908	0.917	0.915	0.914

Table 4.3: Training Time

	v11n	**v11s**	**v11m**	**v11l**	**v11x**
Training (Hours)	0.840	1.176	2.459	3.012	5.354

4.2 Diskusi dan Analisis

Namun, nilai mAP50-95 menunjukkan variasi yang lebih jelas di antara model-model tersebut. Model-model dengan arsitektur v11s dan v11n menunjukkan performa yang lebih baik dalam hal mAP50-95 dibandingkan dengan v11m, v11l, dan v11x. Nilai mAP50-95 yang lebih rendah pada model v11m, v11l, dan v11x menunjukkan bahwa model-model ini memiliki sedikit penurunan dalam mendeteksi objek pada berbagai threshold IoU, yang berarti deteksi pada objek dengan margin toleransi lebih ketat sedikit lebih menurun.

4.2.1 Waktu Pelatihan

Model YOLOv11n membutuhkan waktu pelatihan terpendek, hanya 0.84 jam, sementara model YOLOv11x membutuhkan waktu pelatihan yang jauh lebih lama, yaitu sekitar 5.35 jam. Ini menunjukkan bahwa semakin besar arsitektur model, semakin lama waktu yang dibutuhkan untuk melatih model. Namun, perbedaan waktu ini juga sebanding dengan peningkatan performa deteksi yang diperoleh pada model yang lebih besar.

4.2.2 Analisis Perbandingan Model

Berdasarkan hasil di atas, dapat disimpulkan bahwa model YOLOv11x, meskipun membutuhkan waktu pelatihan yang lebih lama, memberikan hasil terbaik dalam hal mAP50 dan precision, dengan sedikit perbedaan pada mAP50-95. Model ini menunjukkan keunggulan dalam hal deteksi objek secara keseluruhan, meskipun biaya komputasinya lebih tinggi.

Model YOLOv11n menunjukkan kinerja yang cukup baik, dengan waktu pelatihan yang lebih cepat. Namun, performa deteksi pada mAP50-95 sedikit lebih rendah dibandingkan dengan model YOLOv11s, yang lebih seimbang antara akurasi deteksi dan waktu pelatihan.

Model dengan arsitektur YOLOv11m dan YOLOv11l menunjukkan hasil yang relatif seimbang pada metrik precision dan recall, tetapi cenderung menunjukkan performa yang lebih rendah pada mAP50-95, terutama dalam deteksi objek pada IoU yang lebih ketat.

4.2.3 Pengaruh Ukuran Model terhadap Kinerja dan Waktu Pelatihan

Seperti yang terlihat dari hasil eksperimen, ukuran model memiliki dampak yang signifikan terhadap kinerja dan waktu pelatihan. Model dengan kapasitas lebih besar, seperti YOLOv11x, memberikan hasil yang lebih akurat dalam

mendeteksi objek, namun juga membutuhkan waktu pelatihan yang jauh lebih lama. Hal ini menunjukkan trade-off antara keakuratan deteksi dan kebutuhan komputasi. Oleh karena itu, pemilihan model yang tepat harus mempertimbangkan faktor waktu pelatihan dan sumber daya komputasi yang tersedia.

Sementara itu, model dengan kapasitas lebih kecil, seperti YOLOv11n, lebih efisien dalam hal waktu pelatihan, namun dengan sedikit penurunan pada mAP50-95. Ini mungkin menjadi pilihan yang baik untuk aplikasi yang membutuhkan deteksi cepat dengan akurasi yang sangat baik pada threshold IoU yang lebih rendah.

Chapter 5

Penutup

5.0.1 Kesimpulan

Berdasarkan hasil eksperimen, dapat disimpulkan bahwa semua model YOLOv5 (v11n, v11s, v11m, v11l, v11x) memberikan hasil yang sangat baik dalam deteksi objek dengan precision dan recall yang mendekati 1.0. Model YOLOv11x memberikan performa terbaik dalam hal mAP50 dan precision, tetapi membutuhkan waktu pelatihan yang lebih lama. Sementara itu, model YOLOv11n menawarkan waktu pelatihan yang lebih cepat dengan hasil yang cukup baik, terutama pada mAP50.

Pemilihan model harus disesuaikan dengan kebutuhan spesifik aplikasi, di mana aplikasi dengan batasan waktu atau sumber daya komputasi mungkin lebih memilih model dengan kapasitas lebih kecil, sementara aplikasi yang memerlukan akurasi deteksi tinggi dapat memilih model yang lebih besar meskipun dengan biaya komputasi yang lebih tinggi. Dalam konteks deteksi objek seperti Mario, model YOLOv11n saja sudah lebih dari cukup dan bahkan menunjukkan performa yang luar biasa.

5.1 Saran untuk Penelitian Selanjutnya

Untuk penelitian selanjutnya, dapat dieksplorasi penggunaan teknik-teknik optimasi lebih lanjut, seperti quantization atau pruning, untuk mengurangi ukuran model dan waktu pelatihan tanpa mengorbankan akurasi deteksi. Selain itu, penggunaan teknik transfer learning dengan fine-tuning pada dataset yang lebih besar dapat meningkatkan performa model, terutama untuk aplikasi yang memiliki lebih banyak variasi objek yang beragam.

5.2 Refleksi Kelompok

Pada awalnya, kami tidak mengalami kendala signifikan. Namun, saat beralih ke pendekatan deep learning dengan menggunakan model YOLOv11 untuk deteksi

karakter Mario, kami menghadapi beberapa tantangan utama. Kendala utama yang kami temui adalah keterbatasan sumber daya, terutama saat melakukan fitting model. Program sering kali mengalami error karena keterbatasan RAM yang disediakan oleh Kaggle, yang menjadi platform utama kami dalam pengerjaan proyek ini. Selain itu, waktu eksekusi yang memakan waktu sekitar 3-5 jam juga menjadi hambatan karena kami perlu menunggu hasil eksperimen untuk melakukan fine-tuning atau mengeksplorasi lebih dalam berdasarkan hasil prediksi setelah fitting. Untuk mengatasi masalah ini, kami memutuskan untuk menggunakan berbagai mesin dari setiap anggota tim, sehingga eksperimen dapat berlangsung sambil proses fitting tetap berjalan. Selain itu, dataset buatan kami yang digunakan dapat dibilang terbatas karena hanya 570 citra. Keterbatasan ini turut memengaruhi pelatihan model. Kami mencoba untuk mengatasi hal tersebut dengan melakukan augmentasi data untuk meningkatkan performa model dalam deteksi karakter Mario.

Bibliography

Appendix A

Notebook dan Output

Notebook dan output training serta inferensi dapat dilihat pada laman https://www.kaggle.com/code/patricksa
project-cv dimana versi-versinya mengandung model yang berbeda-beda.