BAB I

PENDAHULUAN

1.1. Latar Belakang

Deteksi objek berperan sebagai titik sentral dalam aplikasi krusial seperti manajemen lalu lintas cerdas, pengawasan keamanan video, dan inisiasi infrastruktur kota digital. Fungsi utamanya adalah melokalisasi dan mengklasifikasikan objek spesifik dalam domain visual, sebuah proses yang fundamental dalam menjamin akurasi interpretasi dan pengambilan keputusan yang efektif dalam kondisi operasional yang beragam (Verma dkk., 2024). Secara umum, pendekatan deteksi objek terbagi menjadi dua kategori, yaitu two-stage yang memanfaatkan wilayah deteksi, serta one-stage yang berbasis regresi atau klasifikasi, yang sering dikenal sebagai deteksi objek real-time (Kang dkk., 2022). Aplikasi berbasis deteksi objek real-time, seperti pada teknologi kendaraan otonom, menawarkan manfaat signifikan dalam kehidupan sehari-hari, khususnya dalam meningkatkan keselamatan berkendara. Kendati demikian, salah satu tantangan utama dalam aplikasi ini adalah mendeteksi objek berukuran kecil, seperti tikus atau botol, yang sering kali hanya menempati sebagian kecil dari gambar namun tetap memiliki peran penting dalam analisis visual (Nguyen dkk., 2020).

Pada deteksi objek, dikenal Deep learning yang merupakan jaringan saraf dengan memanfaatkan transformasi dan graf secara bersamaan untuk membangun model pembelajaran berlapis-lapis. (Alzubaidi dkk., 2021). Arsitektur deep learning terdiri dari beberapa hidden layer yang berfungsi mempelajari fitur pada berbagai tingkat abstraksi. Algoritma deep learning dirancang untuk memanfaatkan struktur tersembunyi dalam distribusi input, guna menemukan representasi yang optimal pada berbagai tingkatan. Pada machine learning konvensional, pemrosesan data mentah memerlukan rekayasa fitur manual dan keahlian domain yang mendalam. Pembangunan sistem machine learning mengharuskan adanya ekstraktor fitur yang dapat mengubah data mentah, seperti nilai piksel pada gambar, menjadi vektor fitur yang sesuai agar sistem pembelajaran, seperti pengklasifikasi, dapat mendeteksi pola pada input. Sedangkan pada deep learning, memungkinkan data mentah, seperti piksel gambar, dimasukkan langsung ke dalam algoritma pembelajaran tanpa perlu mengekstraksi fitur secara manual atau mendefinisikan vektor fitur terlebih dahulu (Arif dkk., 2020).

Deteksi objek berbasis deep learning dapat dikategorikan menjadi dua tipe, yaitu taksonomi detektor two-stage dan one-stage. Detektor two-stage mencakup algoritma seperti RCNN (Region-based Convolutional Neural Networks), Faster RCNN, Mask RCNN, Cascade RCNN, FPN (Feature Pyramid Network), dan R-FCN (Region-based Fully Convolutional Networks), sedangkan detektor one-stage meliputi algoritma seperti YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), Refinet, dan RetinaNet (Mittal dkk., 2020). Pada detektor one-stage, algoritma YOLO melakukan klasifikasi objek dengan menggunakan kotak batas

berukuran tertentu yang telah ditentukan pada lapisan tertentu (Lecrosnier dkk., 2021). Detektor two-stage umumnya lebih kompleks dan memiliki kinerja yang lebih unggul dibandingkan detektor one-stage. Namun, YOLO mampu bersaing tidak hanya dengan detektor one-stage, tetapi juga dengan detektor two-stage dalam hal deteksi dan kecepatan inferensi. Dengan arsitektur yang sederhana, rendah kompleksitas, serta mudah diimplementasikan, YOLO menjadi pilihan populer dalam pemodelan deteksi objek (Diwan dkk., 2023). Sebagai detector objek, YOLO membagi gambar menjadi grid berukuran N × N, di mana setiap sel hanya memprediksi satu objek, dengan prediksi berupa sejumlah kotak batas tetap yang masing-masing dilengkapi dengan confidence score. Untuk menghindari prediksi ganda, YOLO menerapkan algoritma non-maxima suppression dalam proses deteksinya. Biasanya, YOLO menggunakan dataset ImageNet untuk pra-pelatihan parameter dan kemudian dilatih dengan dataset target untuk pengenalan objek yang lebih spesifik (Morera dkk., 2020). YOLOv7, salah satu varian terbaru dalam seri YOLO, dirancang dengan fitur trainable bag-of-freebies yang memungkinkan peningkatan kinerja deteksi secara signifikan tanpa menambah biaya inferensi. Selain itu, YOLOv7 menggunakan metode penskalaan ekstend dan compound, yang berfungsi untuk mengurangi jumlah parameter dan perhitungan secara efektif, sehingga secara drastis meningkatkan kecepatan deteksi (Wu dkk., 2022).

YOLOv7 menyediakan berbagai ukuran model, termasuk YOLOv7-Tiny, YOLOv7, YOLOvX, dan YOLOvW. YOLOv7-Tiny mengintegrasikan arsitektur ELAN (Efficient Layer Aggregation Network) untuk ekstraksi fitur. ELAN memperkuat kapasitas pembelajaran jaringan dasar dengan memperluas, mengubah, dan menggabungkan fitur-fitur. Selain itu, ELAN mempercepat konvergensi model melalui jalur gradien yang terkontrol. Penggunaan konvolusi grup memperluas saluran blok komputasi sambil mempertahankan struktur lapisan transformasi. Proses ini meningkatkan kemampuan jaringan backbone dalam mempelajari fitur dan mengoptimalkan penggunaan parameter dalam komputasi (Q. Wang dkk., 2024). Seiring dengan optimalisasi arsitektur, penentuan anchor box juga menjadi aspek penting dalam meningkatkan kemampuan deteksi. Anchor box memainkan peran vital dalam model deteksi objek modern karena menyediakan seperangkat bingkai referensi yang telah ditentukan sebelumnya untuk memfasilitasi pendeteksian objek dengan berbagai ukuran dan rasio aspek dalam suatu gambar (Zhao & Song, 2024). Dalam beberapa penelitian, telah dikembangkan metode adaptif untuk menghasilkan anchor box yang lebih optimal, sehingga mampu meningkatkan kemampuan model dalam mengenali objek, khususnya yang berukuran kecil atau tidak umum.

Objek kecil sendiri lebih sering muncul dibandingkan objek berukuran sedang atau besar, namun keberadaannya sering kali sulit dikenali karena hanya menempati sebagian kecil dari gambar, memiliki kemiripan bentuk atau warna dengan objek yang lain, serta berada dalam latar belakang yang kompleks (Nguyen dkk., 2020). Jumlah piksel yang terbatas pada objek kecil menyebabkan informasi visual yang tersedia menjadi sangat sedikit, sehingga menyulitkan detektor dalam mengenali dan membedakannya secara akurat. Oleh karena itu, dibutuhkan model deteksi objek yang mampu memberikan perhatian khusus terhadap fitur objek kecil. YOLO

merupakan salah satu model deteksi yang unggul dalam kecepatan dan efisiensi komputasi, namun varian ringan seperti YOLOv7-Tiny masih memiliki keterbatasan dalam menangani objek kecil secara optimal. Permasalahan ini semakin kompleks ketika diaplikasikan pada citra jarak jauh, seperti pada sistem berbasis UAV atau kamera pemantau, di mana objek kecil muncul dalam jumlah banyak dan kondisi visual yang bervariasi. Penelitian ini tidak sampai pada tahapan pemasangan model secara langsung ke drone dan berfokus pada perbaikan serta pengujian model menggunakan komputer. Keputusan ini didasarkan pada dua permasalahan. Pertama pada permasalahan hardware, dimana meskipun YOLOv7-Tiny adalah model yang ringan, menjalankannya pada drone tetap saja dapat membebani sistem drone dan membuatnya bekerja terlalu berat. Yang kedua pada permasalahan akurasi model, dimana model yang ringan ini seringkali tidak seakurat model yang lebih besar, terlebih jika diaplikasikan di lingkungan yang rumit (Miao dkk., 2024).

Permasalahan akurasi diperparah karena gambar dari drone yang sulit diolah secara langsung. Gambar hasil drone dapat diambil dari berbagai sudut dan ketinggian. Akibatnya, ukuran objek bisa sangat berbeda-beda, lokasi objeknya bisa menyebar atau berkelompok, dan sering terdapat blur atau gambar buram karena drone dan target sama-sama bergerak sehingga membuat model mudah salah dalam mendeteksi objek (Y. Li dkk., 2024). Kegagalan model saat ini, dalam menangani kondisi tersebut sejalan dengan salah satu penelitian (Miao dkk., 2024) , yang menyatakan bahwa meskipun YOLOv7-tiny sangat efisien untuk platform terbatas sumber daya, model ini masih memiliki kekurangan dalam mendeteksi objek kecil (tiny objects) pada latar belakang yang kompleks. Penerapan model langsung ke dalam drone sangat mungkin dilakukan, yaitu dengan Menerapkan arsitektur Dual-Stage Processing yang membagi beban kerja, dimana drone menggunakan perangkat hemat daya yaitu Raspberry Pi dengan model ringan YOLOv5n untuk navigasi cepat, sementara server darat menggunakan GPU NVIDIA A40 dengan model berat YOLOv8x untuk mengoreksi deteksi via jaringan. Hasilnya, server terbukti mampu mengoreksi kesalahan drone dengan menemukan kembali (re-identify) 70% target yang sempat hilang dengan skor akurasi 0.74. Meskipun unggul dalam menggabungkan kecepatan dan ketelitian, sistem ini memiliki kelemahan berupa ketergantungan pada kualitas sinyal dan hilangnya detail akibat penurunan resolusi citra (downsampling) saat transmisi, yang menyebabkan kegagalan deteksi pada objek yang sangat kecil (Ntousis dkk., 2025).

Penerapan lain pada model ke dalam drone juga dapat dilakukan dengan mengintegrasikan perangkat edge computing seperti NVIDIA Jetson dan menerapkan teknik akselerasi TensorRT guna memangkas waktu pemrosesan. Meskipun pendekatan ini terbukti berhasil meningkatkan Frame Per Second (FPS) secara drastis hingga memenuhi standar real-time, namun hasil penelitian tersebut menunjukkan adanya penurunan kualitas deteksi. Hal tersebut disebabkan karena pemangkasan jumlah parameter pada model versi ringan (Tiny) demi mengejar kecepatan inferensi menyebabkan model kehilangan kedalaman dalam mempelajari fitur objek, yang mengakibatkan nilai Presisi, F1-Score, dan mAP anjlok dibandingkan model versi standarnya (M. Y. Ma dkk., 2023). Mengantisipasi risiko

penurunan akurasi akibat pemangkasan parameter tersebut, penelitian ini tidak berfokus pada implementasi perangkat keras yang rentan menghasilkan deteksi yang kurang akurat. Sebaliknya, penelitian ini memilih pendekatan validasi algoritma (Miao dkk., 2024), yaitu memprioritaskan perbaikan arsitektur algoritmanya terlebih dahulu menggunakan data benchmark (UAV Vehicle Detection dan Flow-Img). Langkah validasi ini penting karena model dasar YOLOv7-Tiny terbukti memiliki kelemahan spesifik pada citra udara, di mana ia sering gagal menangkap fitur objek kecil dan membuang informasi penting yang dibutuhkan untuk deteksi presisi. (Ke dkk., 2025).

Untuk mengatasi permasalahan tersebut, penelitian ini melakukan modifikasi terhadap arsitektur YOLOv7-Tiny dengan menambahkan jalur dan skala deteksi baru, sekaligus memperkuat struktur backbone dan neck guna meningkatkan kemampuan ekstraksi fitur pada resolusi rendah. Selain itu, penentuan anchor box dilakukan secara lebih presisi dengan memanfaatkan Euclidean Distance dan Manhattan Distance dalam proses pemilihan anchor, sehingga distribusi anchor lebih sesuai dengan karakteristik ukuran objek pada dataset. Dengan pendekatan ini, diharapkan model dapat mendeteksi objek kecil secara lebih akurat tanpa mengorbankan kecepatan inferensi, serta dapat diterapkan secara efektif dalam berbagai skenario nyata yang membutuhkan ketelitian tinggi.

1.2. Landasan Teori

1.2.1. Deteksi Objek Kecil dalam Visi Komputer

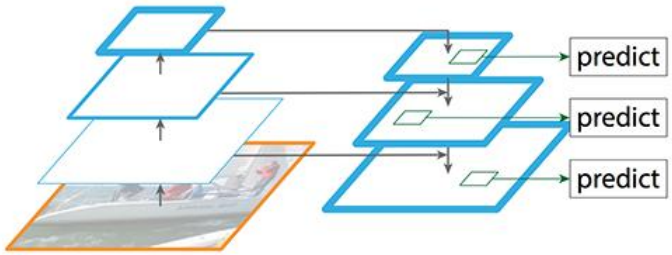
Deteksi objek dalam visi komputer berfungsi melokalisasi dan mengklasifikasikan instansi objek semantik dalam sebuah citra, ditandai dengan kotak batas dan label kelas. Operasi penentuan "apa dan di mana objek itu" dijalankan melalui frontend berbasis Deep Learning yang berfungsi sebagai mesin utama ekstraksi fitur. Akurasi model sangat bergantung pada pelatihan ekstensif menggunakan ribuan data groundtruth yang representatif. Namun, pada deteksi objek yang kecil terdapat permasalahan saat objek dilihat dari platform udara seperti drone. Objek akan cenderung lebih kecil, memiliki resolusi rendah, dan menghadapi berbagai kendala unik, seperti perubahan sudut pandang ekstrem, variasi skala besar, oklusi, dan latar belakang yang kompleks. Faktor-faktor ini membuat identifikasi objek berukuran kecil menjadi salah satu permasalahan mendasar dalam pengembangan sistem visi komputer saat ini (Cao dkk., 2021). Dalam penelitian visi komputer, peneliti telah memberikan perhatian lebih pada pengembangan metode deteksi menggunakan deep learning. Pendekatan ini berupaya meningkatkan kemampuan sistem untuk mengidentifikasi dan menangani objek kecil dengan lebih efektif, meskipun dalam kondisi gambar yang beragam dan dengan resolusi yang bervariasi (Nguyen dkk., 2020).

Istilah "objek kecil" dapat didefinisikan dalam dua cara. Pertama, dapat merujuk pada objek yang secara fisik memiliki ukuran kecil di dunia nyata. Kedua, sesuai dengan metrik evaluasi MS-COCO (Microsoft Common Objects in Context), yaitu objek yang memiliki luas kurang dari atau sama dengan 32 × 32 piksel (Mirzaei

dkk., 2023). Dalam berbagai aplikasi visi komputer, seperti autonomous driving, pencitraan berbasis UAV, dan pengawasan. Kinerja metode deteksi objek kecil masih belum memadai, Banyak metode deteksi objek kesulitan mendeteksi objek kecil karena beberapa kendala, seperti kurangnya informasi dari area objek yang kecil pada gambar, variasi lokasi yang tinggi untuk objek kecil, dan adaptasi metode yang lebih cocok untuk objek berukuran sedang dan besar (Tong dkk., 2020).

1.2.2. Convolutional Neural Network

Secara fundamental, sistem deteksi objek diarahkan untuk menyelesaikan dua tugas utama, yaitu klasifikasi objek dan penentuan posisi objek. Dalam arsitektur CNN, proses deteksi objek dilakukan melalui pembentukan feature map, yaitu representasi yang dihasilkan oleh operasi konvolusi. Feature map ini memiliki bidang reseptif (receptive field) yang menunjukkan luasnya pengaruh area citra asli terhadap representasi fitur di lapisan yang lebih tinggi (Arkin dkk., 2023). Salah satu tantangan dalam deteksi objek yaitu peningkatan deteksi terhadap objek berskala kecil. Peningkatan deteksi pada skala ini dapat diupayakan dengan memperbesar citra masukan atau menurunkan tingkat down-sampling pada CNN untuk mempertahankan resolusi fitur. Namun, pendekatan ini menimbulkan konsekuensi berupa peningkatan biaya komputasi yang substansial. Untuk mengatasi masalah efisiensi tersebut, penelitian terkini mengusulkan konstruksi piramida fitur (feature pyramid) seperti pada Gambar 1 yang memanfaatkan feature map multi-skala dari berbagai lapisan CNN. Strategi ini secara efisien memfasilitasi deteksi objek dengan rentang ukuran yang luas, yaitu objek besar dideteksi melalui fitur tingkat tinggi, sementara objek kecil dideteksi optimal melalui fitur dari lapisan yang lebih rendah. Dengan demikian, paradigma piramida fitur terbukti mampu menjaga keseimbangan antara akurasi deteksi dan efisiensi komputasi tanpa perlu mempertahankan resolusi tinggi pada keseluruhan hierarki jaringan (C. Yang dkk., 2022).



Gambar 1. Piramida Fitur (Feature Pyramid)

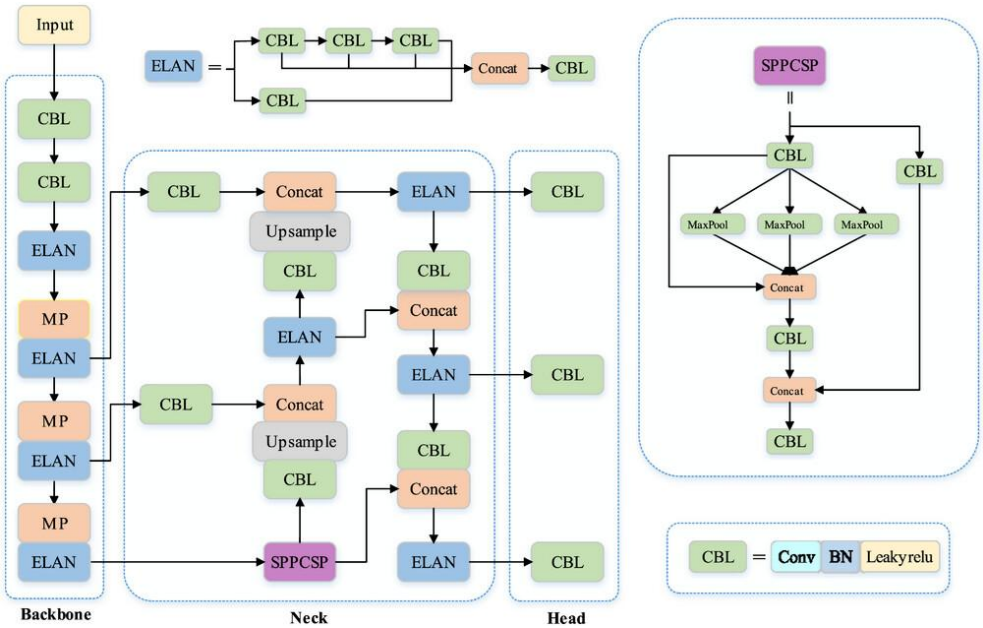
1.2.3. YOLOv7-Tiny

Penelitian ini berfokus pada modifikasi algoritma YOLOv7-tiny, versi teringan dan tercepat dari seri YOLOv7, yang dipilih secara khusus sebagai dasar untuk tugas deteksi objek kecil. Algoritma YOLOv7-tiny memiliki keunggulan berupa kecepatan inferensi yang sangat tinggi dan ukuran model yang ringkas. Meskipun tujuan utama

model tiny adalah implementasi pada perangkat keras dengan sumber daya terbatas, dalam konteks penelitian ini, fokus modifikasi dialihkan pada peningkatan kapabilitas deteksi objek kecil. Pemilihan ini didasarkan pada dua pertimbangan Utama:

a. Efisiensi komputasi yang tinggi dari YOLOv7-tiny memastikan laju pemrosesan yang cepat pada platform GPU standar,
b. Model yang ringan menyediakan platform ideal untuk melakukan modifikasi arsitektural yang terfokus.

Dengan demikian, modifikasi dapat difokuskan untuk meningkatkan sensitivitas dan akurasi model terhadap fitur-fitur halus pada objek berukuran kecil, yang merupakan kelemahan bawaan model ringan (Gong dkk., 2024). Tujuannya adalah menghasilkan model deteksi objek kecil yang akurat namun tetap mempertahankan keunggulan kecepatan pemrosesan yang lebih baik dibandingkan varian YOLOv7 yang lebih besar dan lebih lambat. Model YOLOv7-Tiny meningkatkan jaringan agregasi jarak jauh yang efisien (ELAN) untuk meningkatkan kemampuan deteksi dengan jumlah parameter yang lebih sedikit dan kecepatan deteksi yang lebih tinggi (Z. Yang dkk., 2023). Dalam arsitektur YOLOv7-Tiny terdiri dari empat bagian utama: input, backbone, neck, dan head (C.-Y. Wang dkk., 2022).



Gambar 2. Arsitektur YOLOv7-Tiny

Komponen Input berperan dalam praproses gambar, seperti pemotongan, penskalaan, dan normalisasi, agar sesuai dengan kebutuhan jaringan ekstraksi fitur. Lalu jaringan backbone yang terdiri dari lapisan blok konvolusi atau Convolution Batch Normalization Leaky ReLU (CBL), lapisan Efficient Layer Aggregation Network

(ELAN), dan lapisan Max Pooling (MP). Backbone dirancang untuk mengekstraksi fitur tingkat tinggi dan rendah dari data yang masuk. Blok CBL digunakan untuk mengekstraksi fitur dasar, sedangkan lapisan ELAN mempelajari hubungan antara fitur dasar, dan lapisan MP melakukan concatenation tensor pada fitur dari berbagai skala. Sedangkan pada jaringan neck mengadopsi arsitektur Path Aggregation Feature Pyramid Network (PAFPN) dari seri YOLOv5, yang menggabungkan informasi semantik kuat dari Feature Pyramid Network (FPN) tingkat tinggi dengan tensor informasi posisi kuat dari Path Aggregation Network (PANet) bottom-up. Penggabungan ini secara efektif mengintegrasikan informasi fitur pada berbagai tingkat untuk memungkinkan pembelajaran multi-skala. Dan pada komponen output bertanggung jawab untuk memprediksi tiga peta fitur yang dihasilkan oleh jaringan penggabungan fitur (H. Zhang dkk., 2024).

a. Backbone
   Backbone dalam jaringan saraf YOLOv7-Tiny terdiri dari beberapa modul penting, yaitu CBL, ELAN, MP, dan SPPCSP (Spatial Pyramid Pooling Cross Stage Partial). ELAN, yang mencakup VoVNet dan CSPNet (C.-Y. Wang dkk., 2019), membantu model belajar lebih efektif dengan mengatasi masalah konvergensi dan mengoptimalkan panjang gradien, terutama saat ukuran model diperbesar. Sementara itu, SPPCSP yang merupakan peningkatan dari SPP (Spatial Pyramid Pooling) memungkinkan integrasi peta fitur dari informasi lokal dan global, sehingga memperkaya kemampuan model dalam menangkap detail yang kompleks dan menggabungkan informasi gradien fitur. Dengan demikian, jumlah parameter dapat dikurangi tanpa mengorbankan kinerja, sehingga model menjadi lebih efisien dan efektif dalam melakukan pengenalan atau analisis gambar.
b. Neck
   Neck dalam arsitektur jaringan saraf YOLOv7-Tiny mengadopsi Path-Aggregated Feature Pyramid Network (PAFPN) yang serupa dengan apa yang digunakan pada YOLOv5. PAFPN ini mengintegrasikan Feature Pyramid Network (FPN) dan Path Aggregation Network (PANet), yang menggabungkan peta fitur dari informasi semantik tinggi beresolusi rendah dan informasi semantik rendah beresolusi tinggi melalui proses fusi dari atas ke bawah. Pendekatan ini memperkaya representasi fitur dengan menggabungkan berbagai tingkat resolusi, sekaligus meningkatkan kemampuan model dalam memperoleh informasi lokalisasi yang lebih akurat sepanjang jalur peningkatan. Dengan demikian, neck berperan penting dalam mengoptimalkan performa model dalam deteksi dan pengenalan objek.
c. Head
   Head dalam arsitektur YOLOv7-Tiny memanfaatkan peta fitur dari berbagai skala yang dihasilkan oleh bagian Neck. Setelah melalui tiga lapisan konvolusi standar, bagian ini menghasilkan tiga cabang output yang memproduksi peta fitur dengan skala ukuran untuk deteksi objek berukuran besar, sedang dan kecil. Dengan demikian, model dapat menghasilkan

prediksi akhir yang akurat untuk objek dari berbagai ukuran, meningkatkan kemampuan deteksi dan klasifikasi model secara keseluruhan (Gu dkk., 2023).

Dalam penelitian terkait arsitektur YOLO, beberapa studi telah melakukan modifikasi untuk meningkatkan performa model. Salah satunya (L. Zhang dkk., 2023) adalah penelitian yang mengusulkan algoritma PDWT-YOLO berbasis YOLOv7-tiny, yang dirancang untuk meningkatkan deteksi objek kecil pada citra yang diambil oleh UAV. Modifikasi yang dilakukan meliputi penambahan lapisan deteksi objek kecil, penggantian head deteksi dengan decoupled head guna mengurangi konflik antara tugas klasifikasi dan regresi, serta penerapan loss function Wise Intersection over Union (WIoU) untuk mempercepat konvergensi dan meningkatkan regresi. Berdasarkan pengujian pada dataset VisDrone-2019, PDWT-YOLO menunjukkan peningkatan kinerja dengan mAP@0.5:0.95 naik sebesar 5% dan mAP@0.5 meningkat 6,7% dibandingkan YOLOv7-tiny asli.

Selain itu, penelitian lain (Kim dkk., 2021) juga dilakukan pada arsitektur YOLOv5 dengan fokus pada peningkatan deteksi objek kecil pada citra udara melalui pengembangan ECAP-YOLO. Model ini mengintegrasikan modul Efficient Channel Attention (ECA) dan metode pyramid attention channel, menggantikan konvolusi standar dengan transposed convolution untuk mengurangi kehilangan informasi selama proses upsampling, serta menambahkan lapisan deteksi khusus untuk objek berukuran kecil. Hasil eksperimen menunjukkan peningkatan rata-rata presisi (mAP) sebesar 6,9% pada dataset VEDAI, 5,4% pada kelas mobil kecil di dataset xView, 2,7% pada kelas kendaraan kecil dan kapal kecil di dataset DOTA, serta 2,4% pada kelas mobil kecil di dataset Arirang, jika dibandingkan dengan YOLOv5 asli.

1.2.4. Anchor box

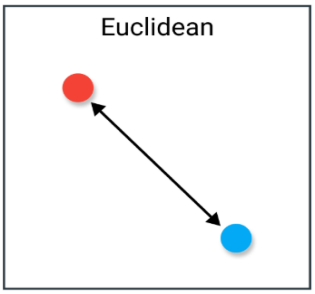
Anchor box adalah sekumpulan kotak referensi dengan ukuran dan rasio aspek yang telah ditentukan, setiap ukuran dan rasio memainkan peran penting dalam model deteksi objek modern seperti YOLO. Anchor box berfungsi sebagai template atau pola awal yang telah ditentukan sebelumnya untuk memfasilitasi pendeteksian objek dengan berbagai ukuran pada bagian head YOLO. Selama proses pelatihan, setiap anchor box akan menjadi terspesialisasi untuk mendeteksi bentuk dan ukuran tertentu. Misalnya, satu anchor untuk objek vertikal besar dan anchor lain untuk objek horizontal kecil (Limberg dkk., 2022). Kinerja YOLO dapat mengalami penurunan saat mendeteksi objek berukuran kecil, terutama dalam kondisi citra yang memiliki tekstur yang kompleks. Salah satu penyebab utama dari penurunan ini adalah penggunaan anchor box sebagai template deteksi, yang umumnya disusun berdasarkan dataset umum seperti COCO (Common Objects in Context). Meskipun mencakup berbagai ukuran objek, anchor box tidak dirancang secara khusus untuk skenario industri dengan objek kecil atau bentuk tidak teratur. Akibatnya, anchor box bawaan sering kali tidak sesuai dengan karakteristik objek spesifik, sehingga mengurangi kemampuan deteksi (Han dkk., 2024). Karena anchor box yang tepat

dapat mengurangi nilai loss dan jumlah perhitungan, serta meningkatkan kecepatan dan akurasi deteksi objek (Liu dkk., 2022). Untuk mengatasi permasalahan tersebut, pemilihan kotak anchor yang optimal sangat penting dalam algoritma deteksi objek satu tahap. Umumnya, kotak anchor dipilih melalui tiga metode, yaitu klasterisasi K-Means, pengaturan manual, atau pelatihan mandiri. Namun, metode ini memiliki keterbatasan dimana K-Means rentan terhadap outlier dan inisialisasi awal, pengaturan manual memerlukan upaya empiris yang signifikan, sementara metode pelatihan tradisional bersifat top-down, statis dan sering menghasilkan ketidaksesuaian antara informasi prior dan distribusi objek aktual (Tai dkk., 2023).

Dalam beberapa penelitian sebelumnya, telah dilakukan pengembangan dengan pendekatan adaptif dalam menghasilkan anchor box yang lebih optimal (Zhao & Song, 2024). Sebagai contoh, pengembangan jaringan pembelajaran adaptif berbasis informasi semantik dilakukan guna menghasilkan anchor box yang disesuaikan secara optimal untuk deteksi kapal, yang terbukti meningkatkan efektivitas segmentasi instance dalam kerangka tugas hibrid berbasis kaskade (T. Zhang & Zhang, 2022). Di sisi lain, metode alternatif yang lebih sederhana dan efisien juga banyak digunakan dalam menentukan anchor box awal, salah satunya adalah algoritma K-means. Algoritma ini melakukan pengelompokan berdasarkan jarak antara data dan pusat klaster (prototipe), sehingga memungkinkan identifikasi representasi bentuk dan ukuran objek yang dominan dalam dataset (Chen dkk., 2020). Dalam penelitian ini, digunakan dua varian K-means berbasis Euclidean distance dan Manhattan distance untuk memperoleh distribusi anchor box yang sesuai dengan karakteristik objek kecil.

a. Euclidean Distance

   Euclidean distance merupakan ukuran jarak lurus antara dua titik dalam ruang, yang dihitung berdasarkan prinsip Teorema Pythagoras. Metode ini menjadi salah satu pendekatan pengukuran jarak yang paling umum digunakan dalam berbagai algoritma pembelajaran mesin (Suwanda dkk., 2020).



Gambar 3. Konsep Euclidean distance dalam ruang dua dimensi

   Euclidean distance yang konsep dasar dalam geometri dan merupakan ukuran spasial yang paling intuitif, menjadikannya operasi matematika inti yang sangat relevan untuk algoritma yang bergantung pada jarak. Dalam algoritma clustering seperti K-Means, perhitungan jarak ini merupakan operasi komputasi yang paling sering dilakukan. Berdasarkan perannya yang fundamental dan umum

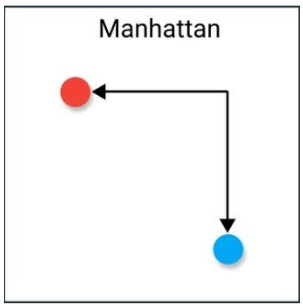
digunakan tersebut, jarak Euclidean dipilih dalam penelitian ini sebagai salah satu metrik utama untuk dievaluasi dalam proses K-Means clustering untuk penentuan anchor box (Mussabayev, 2024). Secara matematis, Euclidean distance diperoleh dengan menghitung akar kuadrat dari jumlah kuadrat selisih antara dua vector atau dengan rumus:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \quad (1)$$

Keterangan:
d_{ij} = jarak antara vektor masukan dan vektor pembanding
n = jumlah vektor
x_{ik} = vektor gambar masukan
x_{jk} = vektor gambar pembanding

b. Manhattan Distance

Berdasarkan literatur, Manhattan distance yang dikenal juga sebagai city block distance atau taxicab distance adalah metode pengukuran jarak yang menghitung total selisih absolut antar atribut dari dua vektor, menjumlahkan perbedaan nilai pada setiap dimensi secara linier (Suwanda dkk., 2020). Berbeda dengan Euclidean distance yang mengukur garis lurus, Manhattan distance menunjukkan akurasi yang lebih tinggi untuk membandingkan perbedaan antara vektor terdekat dan terjauh dari vektor tetap (de Oliveira dkk., 2022). Selain itu, Manhattan distance sangat efektif karena sifatnya yang tahan terhadap nilai outlier dan kesederhanaan perhitungannya, serta kemampuannya untuk menangani vektor fitur berdimensi tinggi dan jarang dengan baik (G. Ma dkk., 2025).



Gambar 4. Konsep Manhattan distance dalam ruang dua dimensi

Untuk dua titik data x_i dan x_j dalam ruang berdimensi-d, Manhattan distance antara keduanya didefinisikan sebagai jumlah dari nilai absolut selisih pada

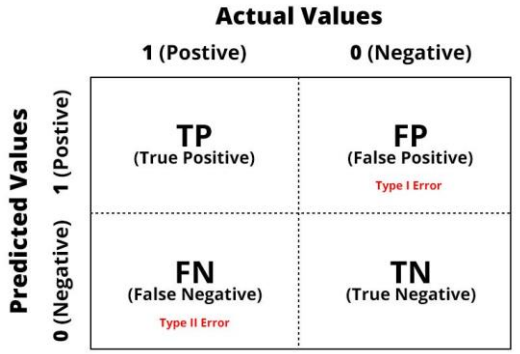
setiap dimensi. Secara matematis, rumus Manhattan distance dinyatakan sebagai:

$$d_{ij} = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \quad (2)$$

Keterangan:
d_{ij} = Manhattan distance antara dua vektor x_i dan x_j
n = jumlah dimensi
x_{ik} = nilai elemen ke-k dari vektor x_i
x_{jk} = nilai elemen ke-k dari vektor x_j

1.2.5. Metrik Evaluasi

Penelitian ini menerapkan sejumlah metrik evaluasi untuk menilai kinerja model yang telah dikembangkan. Metrik yang digunakan meliputi Precision, Recall, F1-Score, Mean Average Precision (mAP), dan Intersection over Union (IoU). Selain itu, evaluasi juga dilengkapi dengan confusion matrix atau Matrixs kesalahan yang menyajikan perbandingan antara hasil prediksi model dengan label sebenarnya dalam bentuk tabel (Hidayat dkk., 2024), sebagaimana ditampilkan pada Gambar 5.



Gambar 5. Confusion Matrix

a. Precision
Precision merupakan rasio antara jumlah prediksi benar positif (True Positive) terhadap total prediksi yang diklasifikasikan sebagai positif, yaitu gabungan antara True Positive (TP) dan False Positive (FP). Nilai ini mencerminkan tingkat ketepatan model dalam menghasilkan prediksi positif yang benar sesuai dengan data sebenarnya (Hidayat dkk., 2024). Untuk menghitung precision, digunakan persamaan (3).

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

b. Recall

Recall merupakan perbandingan antara jumlah prediksi benar positif (True Positive) dengan total data yang seharusnya termasuk dalam kategori positif, yaitu gabungan True Positive (TP) dan False Negative (FN). Metrik ini menggambarkan sejauh mana kemampuan model dalam mendeteksi atau mengidentifikasi seluruh data relevan secara benar (Hidayat dkk., 2024). Untuk menghitung precision, digunakan persamaan (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

c. F1 Score

F1-Score adalah metrik evaluasi yang merepresentasikan keseimbangan antara nilai Precision dan Recall. Metrik ini berguna untuk memberikan gambaran umum mengenai performa model, terutama ketika terdapat ketidakseimbangan antara data positif dan negatif (Teknika dkk., 2024). Perhitungan F1-Score dilakukan berdasarkan rumus yang tercantum dalam Persamaan (5).

$$F1 = 2\,x\,\frac{precision\ x\ recall}{precision + recall} \tag{5}$$

d. Mean Average Precision (mAP)

Mean Average Precision (mAP) merupakan salah satu parameter utama yang digunakan sebagai indikator performa model yang telah dilatih. mAP dihitung dengan mengambil rata-rata nilai Average Precision (AP) dari setiap kelas yang terdapat dalam model. Untuk memperoleh nilai AP pada suatu kelas, diperlukan perhitungan nilai Precision dan Recall terlebih dahulu. Selanjutnya, nilai Precision diplot terhadap nilai Recall, yang umumnya menghasilkan kurva zig-zag. Kurva ini kemudian dihaluskan menggunakan rumus tertentu, sebagaimana ditunjukkan dalam Persamaan (6), untuk memperoleh nilai AP yang lebih representatif (Teknika dkk., 2024).

$$P_{inter}(r_{n+1} - r_n) = \max p(r'); r' \geq r_n + 1 \tag{6}$$

Selanjutnya, nilai AP didapatkan menggunakan persamaan (7).

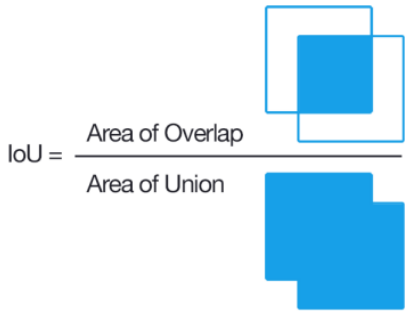
$$AP = \sum (r_{n+1} - r_n) P_{inter}(r_{n+1}) \tag{7}$$

Kemudian untuk mendapatkan nilai mAP dapat dihitung menggunakan persamaan (8).

$$mAP = \sum_{i=1}^{n} \frac{AP(i)}{N} \; x \; 100\% \tag{8}$$

e. Intersection over Union (IoU)
IoU merupakan metrik evaluasi standar yang umum digunakan dalam tugas deteksi objek. IoU berfungsi untuk menilai sejauh mana prediksi bounding box suatu model tumpang tindih dengan ground truth. Metrik ini juga digunakan sebagai dasar untuk menentukan apakah suatu prediksi tergolong True Positive atau False Positive dalam evaluasi performa model (Rezatofighi dkk., 2019).

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{9}$$



Gambar 6. Ilustrasi persamaan IoU

1.3. Perumusan Masalah

Berdasarkan latar belakang masalah di atas, maka rumusan masalah pada penelitian ini yaitu:

1. Bagaimana meningkatkan kemampuan deteksi objek kecil pada YOLOv7-Tiny melalui modifikasi arsitektur?
2. Bagaimana meningkatkan kemampuan deteksi objek kecil pada YOLOv7-Tiny melalui modifikasi anchor box?
3. Sejauh mana kombinasi modifikasi arsitektur dan anchor box dapat meningkatkan kemampuan deteksi objek kecil pada YOLOv7-Tiny?

1.4. Tujuan dan Manfaat

1.4.1. Tujuan

Penelitian ini bertujuan:

1. Mendesain dan mengimplementasikan modifikasi arsitektur YOLOv7-Tiny untuk meningkatkan kemampuan deteksi objek kecil.
2. Melakukan penyesuaian anchor box pada modifikasi YOLOv7-Tiny terhadap distribusi ukuran objek pada dataset.
3. Mengevaluasi dampak kombinasi modifikasi arsitektur dan anchor box pada YOLOv7-Tiny terhadap kemampuan deteksi pada objek kecil.

1.4.2. Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan solusi dalam pengembangan metode deteksi objek berbasis YOLOv7-Tiny dengan fokus pada objek kecil tanpa mengabaikan objek dengan skala lebih besar, melalui modifikasi arsitektur dan anchor box.
2. Menjadi referensi terkait pengaruh metode clustering terhadap penentuan anchor box pada model deteksi berbasis anchor.

1.5. State of The Art

Tabel 1. State of The Art

No.	Judul Penelitian, Penulis dan Tahun Terbit	Objek dan Masalah	Metode Penyelesaian	Hasil Penelitian	Perbedaan Penelitian
1.	Judul : Orientation- and Scale-Invariant Multi-Vehicle Detection and Tracking from Unmanned Aerial Videos Penulis : Jie Wang, Sandra Simeonova dan Mozhdeh Shahbazi Tahun Terbit : 2019	Objek: UAV Vehicle Detection dataset. Masalah: Variasi orientasi dan skala, serta seringnya terjadi pergantian identitas antar objek.	1. Deteksi: Fine-tuning YOLOv3 dengan dataset lalu lintas kustom. 2. Pelacakan: Menggabungkan dua fitur asosiasi data secara berbobot (weighted fusion): Estimasi Gerakan (Filter Kalman) dan Fitur Penampilan Mendalam (Deep Appearance Features / Re-ID) menggunakan Wide Residual Network.	Akurasi Pelacakan (MOTA) mencapai 81.3% dan F1-score Deteksi 92.1%. Berhasil mengurangi ID Switch secara drastis (hanya 1 ID Switch per 305 kendaraan yang dilacak) dan bekerja secara efisien pada 11 fps.	Penelitian tersebut melakukan optimasi deteksi objek dengan fine-tuning pada model YOLOv3. Pada penelitian kami dengan dataset yang sama, dilakukan fine-tuning dengan arsitektur dan anchor YOLOv7-Tiny yang telah dimodifikasi untuk mengatasi keterbatasan YOLOv3 dalam sensitivitas skala kecil dan meningkatkan akurasi deteksi untuk objek dengan resolusi rendah.

No.	Judul Penelitian, Penulis dan Tahun Terbit	Objek dan Masalah	Metode Penyelesaian	Hasil Penelitian	Perbedaan Penelitian
2.	Judul : Improved YOLOv7 for Small Object Detection Algorithm Based on Attention and Dynamic Convolution Penulis : Kai Li, Yanni Wang dan Zhongmian Hu Tahun Terbit : 2023	Objek: FloW-Img sub-dataset. Masalah: Deteksi Objek Kecil pada YOLOv7 baseline mengalami kehilangan fitur, misdeteksi/kebocoran, dan kurang efisien untuk variasi skala objek.	1. SPPFCSPC (Pengganti SPPCSPC untuk kecepatan dan lokalisasi). 2. Coordinate Attention (CA) (Untuk channel dan informasi posisi). 3. Konvolusi Dinamis (Untuk adaptasi kernel terhadap variasi ukuran objek).	Peningkatan signifikan 5.2% mAP dari baseline (mencapai 81.1%) pada dataset objek kecil (FloW-Img), dengan peningkatan biaya komputasi yang minimal (9% FLOPs).	Penelitian tersebut menggunakan YOLOv7 standar untuk performa maksimum. Kami memilih YOLOv7-Tiny yang lebih ringan sebagai bentuk penyesuaian model terhadap kebutuhan real-time dan keterbatasan perangkat. Kontribusi kami adalah melakukan modifikasi arsitektur dan penyesuaian anchor skala baru secara spesifik untuk mengatasi kelemahan YOLOv7-Tiny dalam sensitivitas deteksi objek kecil dan resolusi rendah.

No.	Judul Penelitian, Penulis dan Tahun Terbit	Objek dan Masalah	Metode Penyelesaian	Hasil Penelitian	Perbedaan Penelitian
3.	Judul : Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery Penulis : Linhua Zhang, Ning Xiong, Xinghao Pan, Xiaodong Yue, Peng Wu dan Caiping Guo Tahun Terbit : 2023	Objek: VisDrone-2019 dataset. Masalah: YOLOv7-Tiny dalam pendeteksian objek kecil pada citra UAV dengan latar belakang yang kompleks terlalu banyak mengorbankan akurasi demi kecepatan, sehingga kinerja deteksi objek kecilnya rendah dan membutuhkan peningkatan.	1. Penambahan Lapisan Deteksi P2 untuk mempertahankan fitur objek kecil. 2. Penggunaan Decoupled Head untuk memisahkan tugas Klasifikasi dan Regresi agar mengurangi konflik dan mempercepat konvergensi 3. Penggantian Loss Function: Mengganti CIoU dengan WIoU untuk mengatasi sensitivitas skala dan menyeimbangkan sampel regresi.	Peningkatan Akurasi 6.7% mAP@0.5 (mencapai 41.2%) pada dataset VisDrone, dengan penambahan parameter yang minimal.	Penelitian tersebut mencapai akurasi tinggi dengan menambahkan Decoupled Head dan WIoU Loss pada YOLOv7-Tiny. Kami melakukan modifikasi serupa (Penambahan P2) namun dengan fokus teknis yang berbeda, yaitu mengutamakan penyesuaian anchor skala baru secara eksklusif untuk mengatasi sensitivitas skala dan resolusi rendah pada dataset UAV, tanpa menggunakan Head dan Loss yang lebih kompleks secara komputasi.

No.	Judul Penelitian, Penulis dan Tahun Terbit	Objek dan Masalah	Metode Penyelesaian	Hasil Penelitian	Perbedaan Penelitian
4.	Judul : An optimized YOLO-based object detection model for crop harvesting system Penulis : Mohamad Haniff Junos, Anis Salwa Mohd Khairuddin, Subbiah Thannirmalai dan Mahidzal Dahari1 Tahun Terbit : 2021	Objek: Objek diambil dari perkebunan kelapa sawit (palm oil plantation) di Malaysia. Masalah: Bagaimana mendapatkan model deteksi objek yang memiliki akurasi tinggi sekaligus ringan dan cepat (low computational cost) untuk diterapkan pada sistem panen kelapa sawit otomatis (memperbaiki kelemahan YOLOv3-Tiny).	1. Mengganti Backbone dengan DenseNet201 (untuk fitur kaya). 2. Menerapkan 4 Skala Deteksi (FPN) untuk objek kecil. 3. Mengoptimalkan Anchor Box menggunakan K-Means.	Mencapai mAP tinggi 98.68% dan model yang sangat ringan, menjadikannya solusi akurat dan efisien untuk aplikasi pertanian presisi.	Penelitian tersebut menggunakan YOLOv3-Tiny dengan modifikasi DenseNet201 dan 4 skala deteksi untuk aplikasi pertanian. Penelitian kami menggunakan model yang lebih baru, YOLOv7-Tiny, dan fokus pada modifikasi anchor. Kontribusi kami adalah sistem penyesuaian anchor skala baru menggunakan jarak Euclidean dan Manhattan untuk mengatasi sensitivitas skala objek kecil, tanpa perombakan pada Backbone YOLOv7-Tiny.

Dari Tabel 1, dapat di analisis bahwa terdapat beberapa penelitian terkait deteksi objek kecil dan bentuk efisiensi model dalam penanganan masalah. Berikut penjelasan singkat terkait penelitian dari Tabel 1:

1. Penelitian (J. Wang dkk., 2019) berfokus pada penyelesaian tantangan variasi orientasi dan skala serta isu pergantian identitas (ID Switch) dalam pelacakan kendaraan. Solusi yang diterapkan melibatkan fine-tuning arsitektur YOLOv3 untuk pendeteksian objek yang diperkuat dengan sistem pelacakan berbasis Kalman Filter dan Re-ID (Deep Appearance Features), menghasilkan akurasi pelacakan 81.3% dan efisiensi komputasi yang baik. Penelitian tersebut menunjukkan bahwa fine-tuning merupakan strategi adaptasi model yang fundamental (Zheng dkk., 2025). Penelitian kami juga menerapkan fine-tuning. Namun, perbedaan terletak pada modifikasi struktural arsitektur baseline, yaitu YOLOv7-Tiny. Penelitian kami mengatasi keterbatasan sensitivitas skala deteksi objek kecil dengan mengintegrasikan lapisan skala deteksi tambahan dan melakukan penyesuaian anchor. Modifikasi arsitektur ini merupakan perbaikan yang tidak ditangani oleh fine-tuning pada penelitian sebelumnya.
2. Penelitian kedua (K. Li dkk., 2023) berfokus pada peningkatan YOLOv7 standar untuk deteksi objek kecil dengan fokus pada adaptabilitas fitur melalui Coordinate Attention dan Konvolusi Dinamis, mencapai peningkatan mAP sebesar 5.2%. Penelitian kami memiliki tujuan yang sama, namun memilih baseline YOLOv7-Tiny yang lebih ringan. Pemilihan YOLOv7-Tiny ini didasarkan pada pertimbangan terhadap efisiensi komputasi (Gong dkk., 2024). Kontribusi utama kami adalah modifikasi arsitektur dengan menambahkan skala deteksi objek kecil baru melalui jalur fusi fitur di neck, dilengkapi dengan penyesuaian anchor K-Means berbasis Euclidean dan Manhattan distance. Pendekatan kami secara langsung mengatasi kelemahan struktural tiny model terhadap resolusi rendah, berbeda dengan fokus penelitian kedua pada peningkatan kualitas fitur model yang lebih besar.
3. Penelitian (L. Zhang dkk., 2023) memodifikasi YOLOv7-Tiny untuk dataset VisDrone dengan menambahkan Lapisan Deteksi P2, Decoupled Head, dan WLoU Loss, menghasilkan peningkatan mAP 6.7%. Kami juga melakukan menambahkan Lapisan P2. Namun, dengan fokus yang berbeda, kami memilih untuk mencapai akurasi serupa melalui modifikasi jalur fusi fitur pada neck dan penyesuaian anchor K-Means. Pendekatan ini lebih menekankan pada solusi efisien yang meminimalkan penambahan biaya komputasi.
4. Penelitian selanjutnya (Junos dkk., 2021) berfokus pada deteksi objek di lingkungan perkebunan kelapa sawit, menggunakan YOLOv3 sebagai baseline. Modifikasi dilakukan dengan penggantian backbone dengan DenseNet201 dan penambahan skala deteksi baru, serta mengoptimalkan anchor box menggunakan K-Means. Pendekatan ini berhasil mencapai akurasi sangat tinggi mAP 98.68. Secara garis besar, strategi penambahan skala baru dan penyesuaian anchor box juga kami terapkan. Namun, fokus tersebut diaplikasikan pada pendeteksian objek kecil dengan baseline yang berbeda, yaitu YOLOv7-Tiny. Kami memilih mempertahankan backbone asli model Tiny dan memfokuskan modifikasi pada arsitektur neck dan penyesuaian anchor K-Means dengan mempertimbangkan jarak Euclidean dan Manhattan distance.

BAB II

METODE PENELITIAN

2.1. Tempat dan Waktu

Penelitian ini dimulai pada bulan September 2024 dan berlangsung hingga November 2025. Objek penelitian berupa objek berukuran kecil pada citra dataset UAV dan Flow-Img. Proses training dan testing model dilakukan Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

2.2. Benda Uji dan Alat

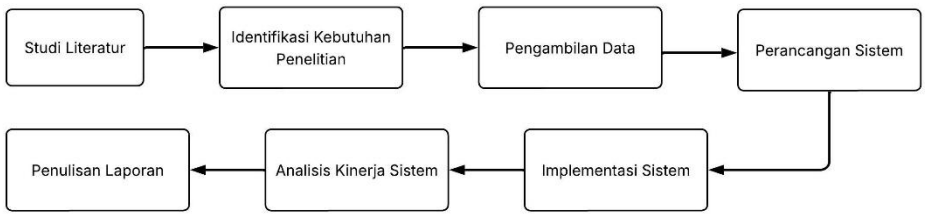
Benda dan alat yang digunakan dalam penelitian ini meliputi perangkat lunak dan perangkat keras berikut:

1. Perangkat lunak
   a. Operating Sistem Operasi Windows 11 Home, 64 bit
   b. Google Colab dengan GPU T4 (NVIDIA Tesla T4 (16 GB VRAM))
   c. Google Drive
   d. Visual Studio Code
   e. *Python*

2. Perangkat keras
   a. Komputer Asus: Dengan prosesor Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz dan RAM 4 GB

2.3. Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahapan, seperti yang ditunjukkan pada Gambar 7:



Gambar 7. Tahapan Penelitian

2.4. Sumber Data

Data yang akan digunakan dalam penelitian ini untuk meninjau dan membandingkan performa deteksi objek adalah UAV Vehicle Detection dataset (J. Wang dkk., 2019) dan FloW-Img sub-dataset (K. Li dkk., 2023).

Tabel 2. Perbandingan Dataset

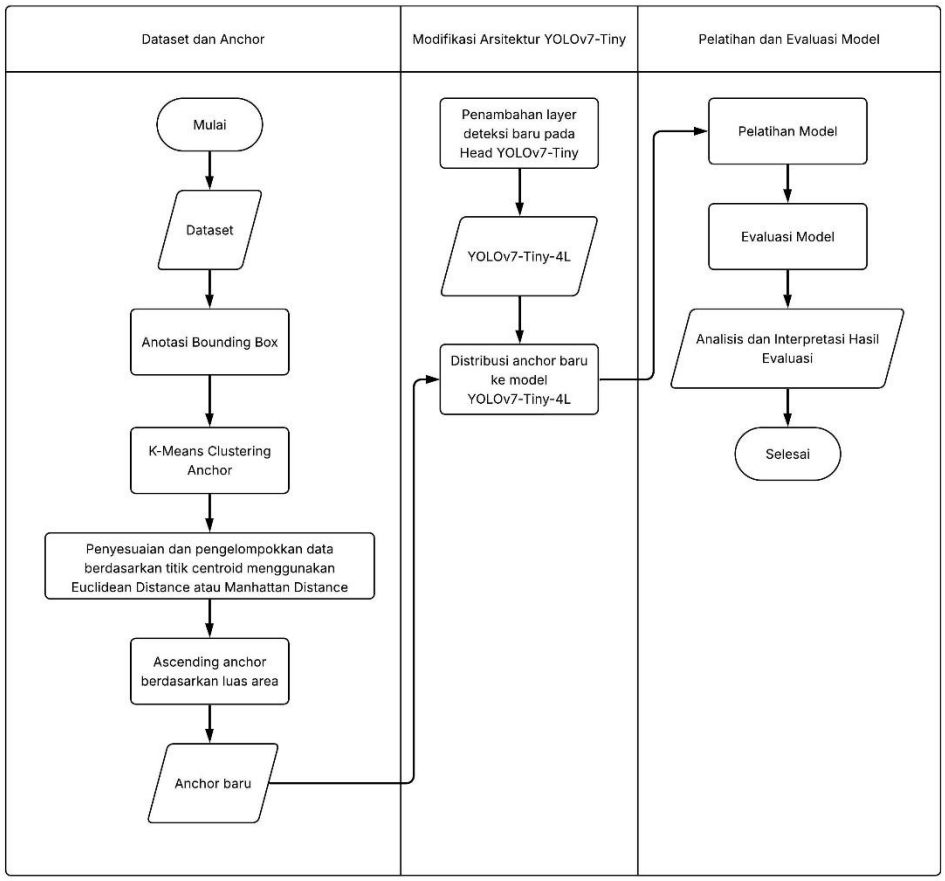
Parameter	UAV Vehicle Detection dataset	FloW-Img sub-dataset
Jumlah Gambar	1.565	2.000
Jumlah Kelas Objek	4	1
Nama Kelas Objek	car, truck, bus dan minibus	bottle
Jumlah Bounding Box	25.183	3.988
Kondisi Pengambilan Gambar	Siang hari dengan cahaya terang	Siang hari dengan cahaya terang dan dengan berbagai kontras
Alat Pengambilan Gambar	Pesawat Tanpa Awak (DJI Phantom 4 Pro, UAVDT-Benchmark, dan FHY-XD-UAV-DATA)	Perahu Tanpa Awak (SMURF Unmanned Cleaning Boat)
Sumber Jurnal	Orientation- and scale-invariant multi-vehicle detection and tracking from unmanned aerial videos	Improved YOLOv7 for Small Object Detection Algorithm Based on Attention and Dynamic Convolution
Sumber Link	https://github.com/jwangjie/UAV-Vehicle-Detection-Dataset	https://orca-tech.cn/en/datasets/FloW/FloW-Img

(a) (b)

Gambar 8. Contoh Gambar a) UAV Vehicle Detection dataset dan b) FloW-Img sub-dataset

2.5. Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran detail mengenai sistem yang akan dikembangkan, serta memahami alur proses dalam sistem tersebut. Adapun alur perancangan sistem usulan arsitektur YOLOv7-Tiny-4L (You Only Look Once versi 7 – Tiny – 4 Layers) ditunjukkan pada Gambar 9.



Gambar 9. Rancangan Sistem

Berdasarkan Gambar 9, perancangan sistem terdiri dari beberapa usulan proses yang diuraikan sebagai berikut:

2.5.1. K-Means Clustering Anchor

Penentuan anchor box merupakan salah satu tahapan dalam menyesuaikan arsitektur YOLOv7-Tiny terhadap distribusi ukuran objek dalam dataset. Anchor box berfungsi sebagai acuan awal dalam proses prediksi bounding box oleh model, sehingga pemilihan anchor yang representatif terhadap ukuran objek pada data dapat meningkatkan kemampuan deteksi, khususnya terhadap objek kecil.

Untuk menentukan anchor box, penelitian ini memanfaatkan algoritma K-Means Clustering guna mengelompokkan dimensi bounding box hasil anotasi dataset. Proses penentuan anchor tersebut dijalankan melalui serangkaian tahapan berikut, sesuai dengan implementasi kode program K-Means Euclidean (terlampir di Lampiran 1) dan K-Means Manhattan (terlampir di Lampiran 2):

1. Ekstraksi Ukuran Bounding Box
   Seluruh bounding box dari data anotasi diekstrak, khususnya lebar (width) dan tinggi (height), yang kemudian dikonversi ke dalam ukuran relatif terhadap dimensi gambar.
2. Penyusunan Dataset Clustering
   Seluruh nilai width dan height disusun ke dalam sebuah array berdimensi $N\ x\ 2$, dimana N adalah jumlah total bounding box dalam dataset. Dataset ini kemudian digunakan sebagai input untuk proses clustering.
3. Penerapan Algoritma K-Means Clustering
   Proses clustering dilakukan menggunakan algoritma K-Means untuk mengelompokkan data bounding box ke dalam sejumlah cluster. Setiap cluster merepresentasikan satu anchor box. Jumlah cluster yang digunakan dalam penelitian ini adalah 12, untuk mendukung tiga prediction head (P2, P3, P4, P5) dengan masing-masing tiga anchor. Proses algortima K-Means terdiri dari Langkah-langkah sebagai berikut:
   a. Inisialisasi centroid awal, yaitu sejumlah 12 titik awal (centroid) dipilih secara acak dari data bunding box.
   b. Pengelompokkan data, yaitu setiap bounding box dihitung jaraknya terhadap seluruh centroid menggunakan metrik jarak. Dalam penelitian ini menggunakan dan membandingkan dua metrik jarak, yaitu Euclidean distance dan Manhattan distance.
   c. Pembaruan centroid, yaitu setiap centroid diperbarui dengan menghitung rata-rata dari seluruh bounding box dalam cluster tersebut, menggunakan rumus:

$$C_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i \tag{10}$$

      Keterangan:
      C_j = centroid baru untuk cluster ke-j
      x_i = vektro bounding box (width, height) ke-i dalam cluster ke-j
      n_j = jumlah anggota cluster ke-j
   d. Iterasi hingga konvergen, yaitu pengelompokkan dan pembaruan centroid dilakukan secara iteratif hingga posisi centroid stabil (konvergen), atau hinggan jumlah iterasi maksimum tercapai.
4. Konversi anchor ke ukuran piksel
   Karena dimensi bounding box yang digunakan masih dalam bentuk relatif, hasil centroid dikalikan dengan ukuran gambar input (416 × 416 piksel) untuk mendapatkan anchor dalam satuan piksel. Nilai anchor kemudian dibulatkan agar sesuai dengan format konfigurasi model YOLO.
5. Penyusunan dan distribusi anchor
   12 anchor yang telah diperoleh kemudian diurutkan berdasarkan luas area (lebar × tinggi), lalu didistribusikan ke 4 prediction head pada model yang di susulkan (YOLOv7-Tiny-4L).

Penelitian ini membandingkan efektivitas dua metrik jarak, Euclidean Distance dan Manhattan Distance, dalam konteks K-Means Clustering. Pada tahap pengelompokan, setiap bounding box dari data anotasi dialokasikan ke cluster berdasarkan centroid terdekat. Pilihan metrik jarak untuk kalkulasi kedekatan ini sangat penting, sebab akan menentukan cara data dikelompokkan serta memengaruhi distribusi dan bentuk akhir dari anchor box. Untuk tujuan perbandingan, kedua metrik tersebut diterapkan secara individual pada dua model yang identik.

1. Euclidean Distance

   Pendekatan ini menghitung jarak antara bounding box dan centroid secara geometris, sehingga hasil cluster yang terbentuk cenderung proporsional dan simetris dengan kode program pada Lampiran 1. Hasil clustering (Tabel 3 dan Tabel 4) menghasilkan 12 anchor box yang dikonversi ke dalam ukuran piksel dengan dimensi gambar 416 × 416 piksel.

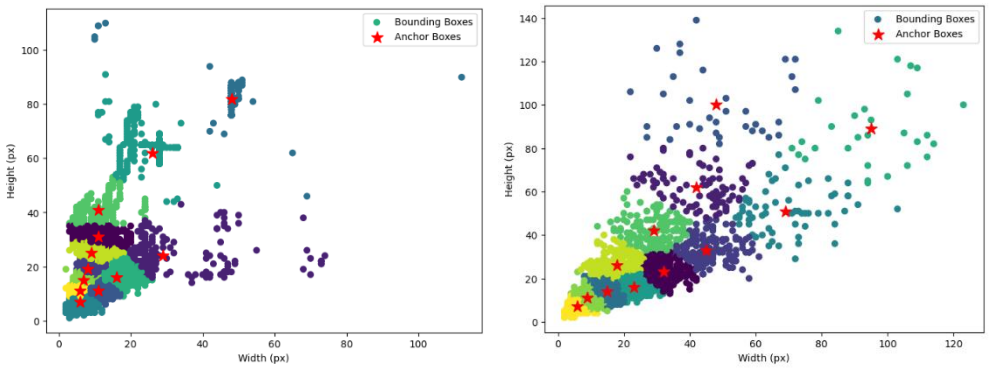
Tabel 3. 12 anchor UAV Vehicle Detection dataset dengan metrik Euclidean distance

Anchor ke-	Width (px)	Height (px)	Area (px^2)
1	6	7	42
2	6	11	66
3	7	15	105
4	11	11	121
5	8	19	152
6	9	25	225
7	16	16	256
8	11	31	341
9	11	41	451
10	29	24	696
11	26	62	1.612
12	48	82	3.936

Tabel 4. 12 anchor Flow-Img dataset dengan metrik Euclidean distance

Anchor ke-	Width (px)	Height (px)	Area (px^2)
1	6	7	42
2	9	11	99
3	15	14	210
4	23	16	368
5	18	26	468
6	32	23	736
7	29	42	1.218
8	45	33	1.485
9	42	62	2.604
10	69	51	3.519
11	48	100	4.800
12	95	89	8.455

Sementara itu, visualisasi hasil clustering ditampilkan pada Gambar 10, yang menunjukkan sebaran bounding box (dalam bentuk titik berwarna) dan posisi anchor yang terbentuk (ditandai dengan simbol bintang merah).



(a) (b)

Gambar 10. Visualisasi Clustering Bounding Box menggunakan Euclidean Distance: (a) UAV Vehicle Detection dataset dan (b) Flow-img dataset

2. Manhattan Distance

Pada pendekatan kedua, digunakan Manhattan Distance untuk mengukur jarak antar data dan centroid. Tidak seperti Euclidean, Manhattan Distance menghitung jarak berdasarkan jumlah total selisih antar dimensi bounding box, sehingga lebih toleran terhadap objek dengan dimensi yang tidak seimbang. Pendekatan ini menghasilkan anchor yang lebih menyebar dan mampu mencakup objek-objek kecil yang sulit ditangkap oleh Euclidean Distance. Berikut daftar 12 anchor pada masing-masing dataset yang digunakan:

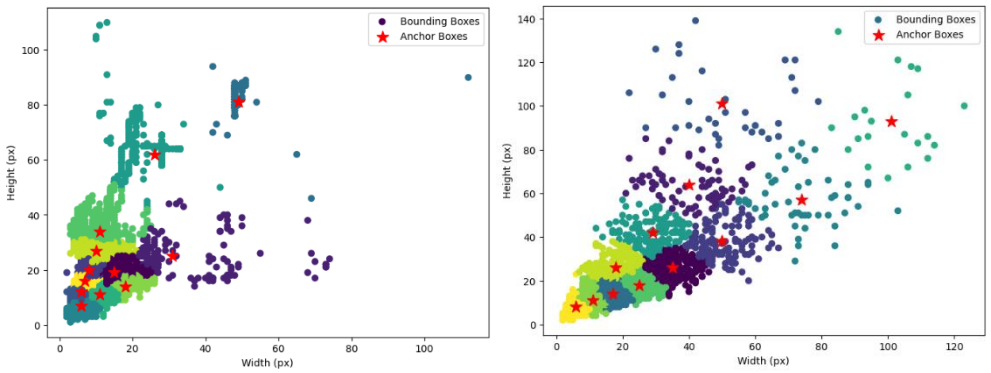
Tabel 5. 12 anchor UAV Vehicle Detection dataset dengan metrik Manhattan distance

Anchor ke-	Width (px)	Height (px)	Area (px^2)
1	6	7	42
2	6	12	72
3	7	16	112
4	11	11	121
5	8	20	160
6	18	14	252
7	10	27	270
8	15	19	285
9	11	34	374
10	31	25	775
11	26	62	1.612
12	49	81	3.969

Tabel 6. 12 anchor Flow-Img dataset dengan metrik Manhattan distance

Anchor ke-	Width (px)	Height (px)	Area (px^2)
1	6	8	48
2	11	11	121
3	17	14	238
4	25	18	504
5	18	26	468
6	35	26	910
7	29	42	1.218
8	50	38	1.900
9	40	64	2.560
10	74	57	4.218
11	50	101	5.050
12	101	93	9.393

Visualisasi hasil clustering ditampilkan pada Gambar 11, terlihat bahwa distribusi anchor lebih menyebar ke berbagai ukuran bounding box. Hal ini menunjukkan keunggulan Manhattan Distance dalam menangkap keragaman dimensi objek.



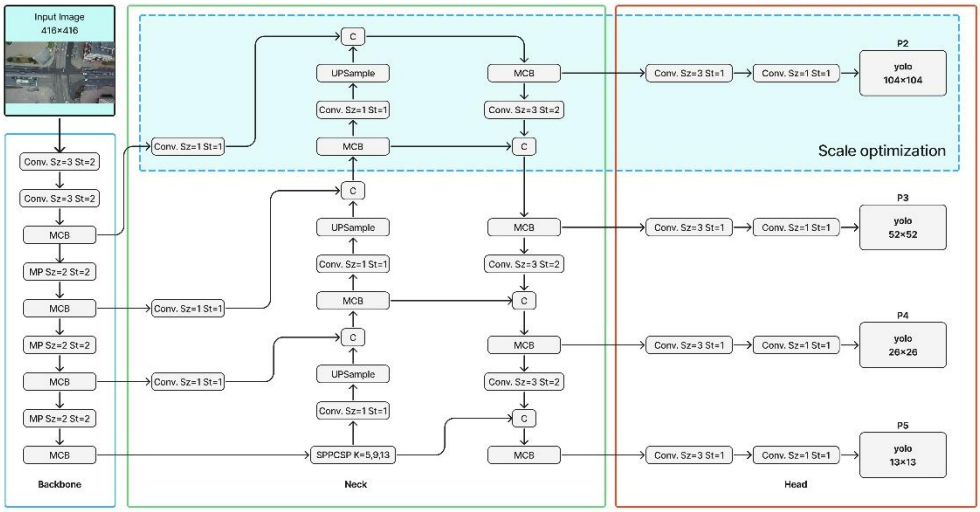
(a) (b)

Gambar 11. Visualisasi Clustering Bounding Box menggunakan Manhattan Distance: (a) UAV Vehicle Detection dataset dan (b) Flow-img dataset

2.5.2. Arsitektur YOLOv7-Tiny

Penelitian ini menggunakan model YOLOv7-Tiny sebagai dasar pengembangan sistem deteksi objek, dengan fokus pada peningkatan performa dalam mendeteksi objek kecil. YOLOv7-Tiny dipilih karena memiliki struktur yang ringan dan efisien, sehingga cocok untuk diimplementasikan pada perangkat dengan sumber daya terbatas. Untuk meningkatkan kemampuan deteksi terhadap objek kecil, dilakukan beberapa modifikasi arsitektur pada YOLOv7-Tiny menjadi YOLOv7-Tiny-4L.

Modifikasi ini mencakup perubahan pada bagian backbone, neck, dan head, dengan penambahan jalur deteksi serta pengolahan fitur resolusi tinggi. Arsitektur hasil modifikasi dapat dilihat pada Gambar 12.



Gambar 12. Arsitektur YOLOv7-Tiny-4L

Kode program YOLOv7-Tiny-4L secara total memiliki 107 lapisan untuk komponen backbone dan neck. Modifikasi spesifik pada model ini terletak pada lapisan ke-68 sampai ke-88 (Lampiran 4). Sementara itu, komponen head tersusun atas 16 lapisan. Jumlah tersebut sudah termasuk implementasi lapisan skala deteksi tambahan (P2) di awal head, dengan konfigurasi 4 lapisan program untuk setiap skala deteksi (Lampiran 5). Adapun modifikasi yang dilakukan dijelaskan sebagai berikut:

1. Penambahan Skala Deteksi Tambahan (P2)
   YOLOv7-Tiny standar hanya menggunakan tiga skala deteksi (P3, P4, dan P5). Dalam penelitian ini, ditambahkan satu skala tambahan yaitu P2 (104×104) dengan stride 4, yang dirancang untuk mendeteksi objek berukuran kecil. Penambahan skala ini dilakukan dengan mengambil fitur dari tahap awal backbone dan mengarahkannya ke bagian neck melalui jalur tambahan untuk diproses lebih lanjut.
2. Modifikasi pada Bagian Neck
   Untuk mendukung skala P2, dilakukan penambahan jalur baru pada bagian neck yang terdiri dari beberapa lapisan, yaitu:
   a. Convolutional layer (Conv)
   b. Modified Convolution Block (MCB)
   c. Upsampling
   d. Concatenation (C)

   Jalur ini dirancang untuk mengolah fitur resolusi tinggi dan menggabungkannya dengan fitur dari layer lain, sehingga dapat

meningkatkan kualitas informasi spasial yang dibutuhkan untuk mendeteksi objek kecil. Selain itu, bagian neck juga dilengkapi dengan blok SPPCSP untuk memperkaya informasi fitur dengan konteks spasial multi-skala.

3. Penyesuaian Head Deteksi
   Jumlah YOLO head ditambah menjadi empat, dengan masing-masing head bertugas pada skala berikut:
   a. P2: 104×104 (stride 4)
   b. P3: 52×52 (stride 8)
   c. P4: 26×26 (stride 16)
   d. P5: 13×13 (stride 32)

   Setiap head memproses fitur pada skala tertentu untuk menghasilkan prediksi kelas, posisi bounding box, dan confidence score. Penyesuaian ini dilakukan agar model mampu mendeteksi objek dalam berbagai ukuran, khususnya objek kecil.

Adapun rincian input dan output dari masing-masing layer pada arsitektur modifikasi YOLOv7-Tiny-4L:

Tabel 7. Rincian Output dari Lapisan-lapisan Arsitektur YOLOv7-Tiny-4L

Bagian Arsitektur	Lapisan	Patch Size/Stride	Ukuran Keluaran	Filter
Backbone	Conv	3x3/2	208x208	32
	Conv	3x3/2	104x104	64
	MCB/ELAN-T	1x1/1 3x3/1	104x104	64
	MP	2x2/2	52x52	
	MCB/ELAN-T	1x1/1 3x3/1	52x52	128
	MP	2x2/2	26x26	
	MCB/ELAN-T	1x1/1 3x3/1	26x26	256
	MP	2x2/2	13x13	
	MCB/ELAN-T	1x1/1 3x3/1	13x13	512
Neck	SPPCSP	1x1/1 5x5/1 9x9/1 13x13/1	13x13	256
	Conv	1x1/1	13x13	128
	Upsample	1x1/2	26x26	128
	Concate	1x1/1	26x26	256

	MCB/ELAN-T	1x1/1 3x3/1	26x26	128
	Conv	1x1/1	26x26	64
	Upsample	1x1/2	52x52	64
	Concate	1x1/1	52x52	128
	MCB/ELAN-T	1x1/1 3x3/1	52x52	64
	Conv	1x1/1	52x52	32
	Upsample	1x1/2	104x104	32
	Concate	1x1/1	104x104	64
	MCB/ELAN-T	1x1/1 3x3/1	104x104	32
	Conv	3x3/2	52x52	64
	Concate	1x1/1	52x52	128
	MCB/ELAN-T	1x1/1 3x3/1	52x52	64
	Conv	1x1/1	26x26	128
	Concate	1x1/1	26x26	256
	MCB/ELAN-T	1x1/1 3x3/1	26x26	128
	Conv	1x1/1	13x13	256
	Concate	1x1/1	13x13	512
	MCB/ELAN-T	1x1/1 3x3/1	13x13	256
Head	Conv	3x3/1	104x104	64
	Conv	1x1/1	104x104	27
	Yolo P2			
	Conv	3x3/1	52x52	128
	Conv	1x1/1	52x52	27
	Yolo P3			
	Conv	3x3/1	26x26	256
	Conv	1x1/1	26x26	27
	Yolo P4			
	Conv	3x3/1	13x13	512
	Conv	1x1/1	13x13	27
	Yolo P5			