



Analisis Eksperimental Penggantian Backbone YOLOv5 dengan EfficientNet-B4 pada Sistem Deteksi Nilai Mata Uang

Alfrida Sabar^{1*}, Ema Utami¹

¹Program Pascasarjana, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

*Corresponding Author's e-mail: alfrida-sabar@students.amikom.ac.id

Article History:

Received: January 6, 2026
Revised: January 16, 2026
Accepted: January 26, 2026

Keywords:

YOLOv5, EfficientNet-B4,
Currency Detection,
Currency Recognition

Abstract: YOLOv5 is a variant of the You Only Look Once (YOLO) algorithm, which is widely recognized for its ability to perform fast and efficient object detection in a single-stage processing framework. The performance of YOLOv5 is strongly influenced by its backbone architecture, which is responsible for extracting visual features from input images. Consequently, backbone replacement is commonly employed as an experimental approach to analyze the impact of feature extraction variations on object detection performance. This study aims to conduct an experimental evaluation of replacing the YOLOv5 backbone with the EfficientNet architecture, specifically EfficientNet-B4, for a rupiah currency value detection system. The experiment was carried out by comparing the baseline YOLOv5 model with a modified YOLOv5 model incorporating the EfficientNet-B4 backbone, using a rupiah banknote dataset consisting of seven nominal classes. Model performance was evaluated using precision, recall, $mAP@0.5$, $mAP@0.5:0.95$, and inference time as evaluation metrics. The experimental results indicate that the use of EfficientNet-B4 leads to a decrease of 1.075% in $mAP@0.5$, while simultaneously increasing $mAP@0.5:0.95$ by 2.247%. This improvement suggests enhanced bounding box localization accuracy under more stringent Intersection over Union (IoU) evaluation criteria. However, the inference time increased significantly, from approximately 7 ms to 37.8 ms per image. Overall, these findings indicate that backbone replacement provides different performance characteristics and needs to be tailored to the specific requirements of the target application.

Copyright © 2026, The Author(s).

This is an open access article under the CC-BY-SA license



How to cite: Sabar, A., & Utami, E. (2026). Analisis Eksperimental Penggantian Backbone YOLOv5 dengan EfficientNet-B4 pada Sistem Deteksi Nilai Mata Uang. *SENTRI: Jurnal Riset Ilmiah*, 5(1), 736–758. <https://doi.org/10.55681/sentri.v5i1.5611>

PENDAHULUAN

Perkembangan teknologi Artificial Intelligence (AI) telah membawa perubahan besar dalam cara sistem pengenalan dan deteksi objek bekerja, termasuk deteksi nilai mata uang. AI memungkinkan komputer untuk memahami pola-pola yang sulit dikenali, seperti perbedaan halus pada angka, warna, atau desain pada uang kertas. Teknologi ini juga mampu beradaptasi dengan berbagai kondisi, seperti cahaya yang redup, sudut pandang yang berbeda, atau kondisi uang kertas yang telah usang. Sistem berbasis AI mampu mengatasi tantangan dalam pengenalan uang kertas, termasuk dalam berbagai kondisi fisik seperti lipatan, rusak serta di bawah pencahayaan yang beragam, sehingga memastikan akurasi dan keandalan yang tinggi dalam penggunaannya (Rafi et al, 2024).

Saat ini banyak penelitian telah mengembangkan cara-cara mendeteksi objek dan banyak dari penelitian tersebut merekomendasikan penggunaan algoritma YOLO untuk deteksi objek secara langsung. Algoritma YOLO dianggap memiliki akurasi deteksi yang

tinggi dan kecepatan yang lebih baik dibandingkan metode lain, sehingga cocok untuk mendeteksi objek secara real-time. Algoritma ini pertama kali diperkenalkan oleh Joseph Redmon et al, dalam penelitiannya menemukan metode untuk mendeteksi objek secara langsung (Redmon et al, 2016). Dalam penelitian tersebut, YOLO terbukti mampu mendeteksi objek dengan kecepatan dan akurasi yang sangat tinggi. YOLO memproses seluruh gambar hanya dalam satu kali proses atau sekali umpan (*single shot*) untuk mendeteksi semua objek sekaligus, dan mampu memberikan hasil deteksi objek secara *real-time*. Namun algoritma ini pun memiliki kendala seperti false positives dan false negatives saat mendeteksi objek kecil (Xu et al, 2024).

Algoritma YOLO terdiri dari tiga komponen utama yaitu backbone, neck dan head dimana masing-masing memiliki fungsi yang berbeda. Backbone adalah jaringan saraf konvolusional yang digunakan untuk mengambil gambar input dan mengekstrak fitur dari gambar input. Ini adalah komponen krusial dalam mendeteksi objek apa pun, karena merupakan kerangka kerja utama yang bertanggung jawab untuk mengekstrak informasi kontekstual dari gambar masukan dan mengubahnya menjadi pola yang lebih abstrak (Nugroho et al., 2021). Beberapa penelitian sebelumnya telah melakukan penggantian backbone pada arsitektur YOLO menggunakan berbagai jaringan konvolusional, seperti EfficientNet, MobileNet, ResNet, dan DenseNet. Tujuannya untuk menyelidiki dampak berbagai backbone terhadap kinerja YOLO sebagai model deteksi objek.

Meena et al dalam penelitiannya menjelaskan bahwa melalui pendekatan hybrid dengan menggabungkan empat model berbeda yaitu Convolutional Neural Network (CNN) yang berbeda yaitu Single-shot Detector (SSD), You Only Look Once (YOLO), RetinaNet, dan Faster R-CNN tercapai akurasi yang lebih baik dalam pengenalan Multi-Label Object dibandingkan dengan metode tradisional (Meena et al, 2023). Hybrid Neural Network Architecture merupakan pendekatan arsitektur jaringan saraf tiruan yang menggabungkan berbagai jenis model atau teknik pembelajaran mesin untuk mencapai kinerja yang lebih baik. Arsitektur hybrid ini menggabungkan keunggulan dari beberapa jenis jaringan neural atau algoritma lain untuk mengatasi keterbatasan dari satu model tunggal.

Dalam kasus deteksi nilai mata uang, arsitektur yang digunakan untuk mengganti backbone asli YOLOv5 adalah arsitektur EfficientNet-B4 yang memiliki keunggulan kemampuan ekstraksi fitur visual yang lebih kaya dan representatif melalui prinsip *compound scaling*, sehingga berpotensi menangkap detail visual uang kertas secara lebih baik, terutama pada aspek tekstur, pola, dan elemen desain yang kompleks. Penelitian ini bertujuan untuk mengevaluasi secara eksperimental penggunaan arsitektur EfficientNet-B4 sebagai backbone pengganti backbone default YOLOv5 (CSPDarknet) dalam sistem deteksi nilai mata uang rupiah, dengan meninjau kinerja deteksi, ketelitian lokalisasi, serta efisiensi komputasi.

Meskipun YOLOv5 telah banyak dikembangkan melalui penggantian arsitektur backbone untuk meningkatkan performa deteksi objek, kajian yang secara sistematis mengevaluasi penggunaan arsitektur EfficientNet sebagai backbone alternatif pada YOLOv5 dalam konteks deteksi nilai mata uang rupiah masih terbatas. Selain itu, penelitian terdahulu belum secara eksplisit membahas perbandingan kinerja antara beberapa varian EfficientNet dengan tingkat kompleksitas yang berbeda, seperti EfficientNet-B0, EfficientNet-B1, dan EfficientNet-B4, serta ketelitian lokalisasi objek berdasarkan metrik evaluasi yang lebih komprehensif, seperti $mAP@0.5:0.95$. Oleh karena itu, penelitian ini menyajikan evaluasi eksperimental penggunaan EfficientNet-B0,

EfficientNet-B1, dan EfficientNet-B4 sebagai backbone YOLOv5 pada sistem deteksi nilai mata uang rupiah, guna memberikan pemahaman metodologis mengenai pengaruh variasi arsitektur backbone terhadap kinerja sistem deteksi.

Ke depannya, sistem ini diharapkan dapat dikembangkan dan diterapkan pada berbagai perangkat, seperti alat bantu deteksi nilai mata uang untuk penyandang tuna netra, mesin kasir otomatis, mesin teller atau ATM, layanan penukaran uang (*money changer*) untuk wisatawan, serta alat deteksi uang palsu. Dengan begitu, sistem ini dapat membantu mempermudah transaksi keuangan, meningkatkan keamanan, mendukung berbagai kebutuhan masyarakat, dan membantu mengurangi peredaran uang palsu. Adapun fokus penelitian ini adalah:

1. Bagaimana implementasi algoritma YOLOv5 dalam mendeteksi nilai mata uang kertas rupiah.
2. Bagaimana implementasi kombinasi algoritma YOLOv5 dan arsitektur EfficientNet-B4 dalam mendeteksi nilai mata uang kertas rupiah.

LANDASAN TEORI

Uang kertas rupiah adalah alat pembayaran yang sah di Indonesia yang diterbitkan oleh Bank Indonesia (BI). Uang kertas rupiah terdiri atas beberapa nilai nominal rupiah yang dilengkapi dengan detail dan fitur spesifik yang berbeda untuk masing-masing nilai mata uang. Berikut adalah pecahan uang kertas rupiah tahun emisi 2022 disertai detail objeknya:

- a. Rp. 1.000: Pahlawan Tjut Meutia, Tari Tifa, Banda Neira dan Bunga Anggrek Larat
- b. Rp. 2.000: Pahlawan M. Husni Thamrin, Tari Piring, Ngarai Sianok, dan Bunga Jeumpa
- c. Rp. 5.000: Pahlawan DR. K. H. Idham Chalid, Tari Gambyong, Gunung Bromo dan Bunga Sedap Malam
- d. Rp. 10.000: Pahlawan Frans Kaisiepo, Tari Pakarena, Taman Nasional Wakatobi, dan Bunga Cempaka Hutan Kasar
- e. Rp. 20.000: Pahlawan DR. G.S.S.J Ratulangi, Tari Gong, Derawan, dan Bunga Anggrek Hitam.
- f. Rp. 50.000: Pahlawan Ir. H. Djuanda Kartawidjaja, Tari Legong, Taman Nasional Komodo, dan Bunga Jepun Bali.
- g. Rp. 100.000: Pahlawan Ir. Soekarno dan Drs. Mohammad Hatta, Topeng Betawi, Raja Ampat dan Bunga Anggrek.

Selain warna, detail serta fitur spesifik pada setiap pecahan uang kertas rupiah dimanfaatkan dalam sistem deteksi objek untuk meningkatkan akurasi dalam mengenali dan mengklasifikasikan masing-masing pecahan secara tepat.



Gambar 1. Gambar uang kertas rupiah

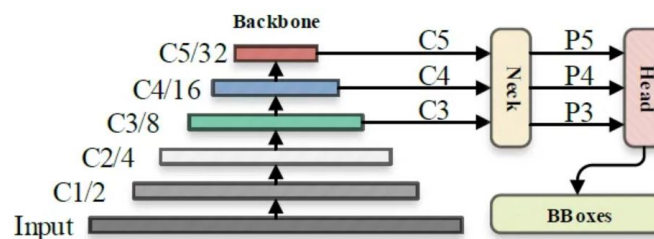
Hingga saat ini, penelitian tentang pengenalan mata uang terus berkembang pesat, dengan berbagai algoritma dan pendekatan yang diuji untuk mencapai performa prediksi gambar yang lebih tinggi. Para peneliti dari berbagai negara telah mengeksplorasi beragam teknik *machine learning* dan *deep learning*, mulai dari metode klasik seperti Support Vector Machines (SVM) hingga arsitektur neural network yang lebih canggih seperti Convolutional Neural Networks (CNN) dan variasinya. Setiap pendekatan memiliki kelebihan dan tantangannya masing-masing, dengan fokus pada peningkatan kinerja model, kecepatan pemrosesan, dan kemampuan generalisasi terhadap berbagai kondisi pengambilan gambar. Perkembangan teknik tersebut kemudian mendorong munculnya model-model Hybrid Neural Network Architecture yang memadukan model YOLOv5 dengan arsitektur modern lain seperti EfficientNet yang diharapkan dapat meningkatkan performa deteksi objek.

Model YOLOv5

YOLOv5 atau "You Only Look Once" versi ke-5 merupakan generasi lanjutan dari model YOLO yang dirancang untuk memberikan performa lebih baik dalam mendeteksi objek secara *real-time* dengan akurasi tinggi dan kecepatan pemrosesan yang efisien. Arsitektur YOLOv5 dikembangkan dengan desain yang fleksibel dan ringan, sehingga dapat digunakan pada perangkat keras dengan sumber daya terbatas seperti smartphone dan IoT devices.

Arsitektur YOLOv5 terdiri dari tiga komponen utama, yaitu Backbone, Neck, dan Head, yang masing-masing memiliki peran penting dalam mendeteksi objek pada gambar atau video. Pada YOLOv5, backbone yang digunakan adalah CSPDarknet53. Backbone berperan sebagai bagian dari convolutional neural network yang bertugas mengekstraksi fitur-fitur penting dalam gambar dan diolah secara bertahap. CSPDarknet53 berperan dalam memisahkan dan menggabungkan informasi gradien secara iteratif, serta mengintegrasikan perubahan gradien ke dalam peta fitur. Hal ini berdampak pada peningkatan akurasi dan efisiensi model serta mengurangi ukuran model dengan mengurangi jumlah parameter (Sun et al, 2024).

Neck terletak di tengah-tengah backbone dan head bertindak sebagai penghubung antara keduanya (Ery Pamungkas dkk, 2022). Neck bertugas menggabungkan fitur dari berbagai tingkatan untuk memastikan deteksi yang akurat. Sedangkan Head terdiri dari tiga cabang, masing-masing berfungsi untuk memprediksi fitur pada skala yang berbeda. Setiap cabang menghasilkan *bounding box*, *class probability*, and *confidence score*. Proses mengubah gambar input menjadi *bounding box* disebut proses inferensi. Gambar 2 berikut merupakan arsitektur YOLOv5 dapat menjelaskan proses inferensi dari YOLOv5 (Liu et al, 2022).



Gambar 2. Arsitektur YOLOv5

Dalam hal deteksi nilai mata uang, melalui Gambar 2 dijelaskan bagaimana prosesnya dimulai dari Input berupa gambar uang kertas rupiah dimasukkan ke dalam

sistem model YOLOv5 dalam resolusi 640×640 piksel. Sistem akan membaca elemen-elemen penting seperti angka nominal, warna, dan fitur keamanan. Backbone akan menerima gambar dan bertugas mengekstraksi fitur utama dari gambar menggunakan CSPDarknet53. Beberapa layer (C1, C2, C3, C4, C5) menunjukkan berbagai tahap ekstraksi fitur:

- Layer C1/C2 menangkap informasi dasar seperti warna dan tekstur.
- Layer C3/C4 mulai mengenali angka nominal, logo Bank Indonesia dan tanda keamanan.
- Layer C5 mengekstraksi fitur tingkat tinggi seperti kombinasi pola unik pada mata uang kertas.

Setelah fitur penting diekstraksi oleh backbone, fitur-fitur ini diteruskan ke bagian Neck yang menggunakan *Feature Pyramid Network* (FPN) dan PANet untuk menggabungkan fitur dari level tinggi (C5), menengah (C4), dan rendah (C3). Di bagian ini, fitur yang diekstraksi dikombinasikan dan diringkas lebih lanjut agar sistem bisa melihat objek dari berbagai skala atau ukuran. P3, P4, dan P5 adalah hasil dari proses ini, dimana model telah mengenali objek pada skala yang berbeda-beda. Pada tahap ini model mengenali uang kertas dalam berbagai skala dan posisi serta memastikan model bisa membedakan nominal uang meskipun warnanya mirip.

Pada bagian Head P3, P4, P5 bertanggung jawab dalam membuat prediksi akhir yaitu:

- *Bounding box* berfungsi menunjukkan kemungkinan lokasi uang kertas dalam gambar.

$$B = (x,y,w,h) \quad (1)$$

Dimana:

x,y : Koordinat pusat objek relatif terhadap grid.

w,h : Lebar dan tinggi kotak pembatas relatif terhadap ukuran gambar

- *Confidence score* untuk menentukan tingkat kepastian model dalam identifikasi tersebut. Nilai *Confidence score* berkisar antara 0%-100%. Jika *Confidence score* rendah, bounding box tersebut kemungkinan salah deteksi (false positive) dan bisa dihapus.

$$\text{Confidence} = P(\text{Object}) \cdot \text{IOU}_{\text{pred}}^{\text{truth}} \quad (2)$$

Dimana:

$P(\text{Object})$: Probabilitas ada objek dalam grid.

$\text{IOU}_{\text{pred}}^{\text{truth}}$: Intersection Over Union antara bounding box prediksi dan ground truth (ukuran seberapa akurat prediksi).

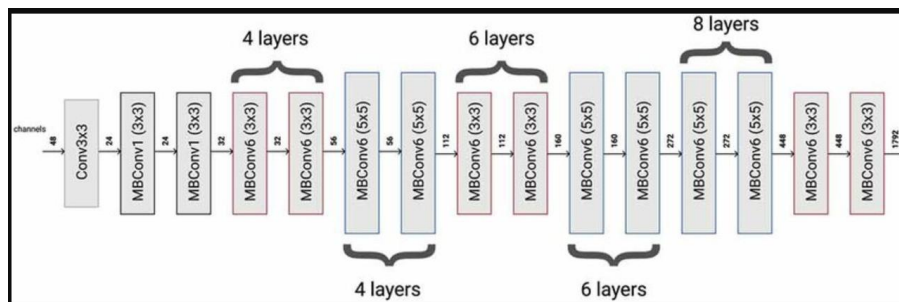
- *Class Probability* untuk mengidentifikasi denominasi uang dengan menghitung probabilitas untuk setiap kategori yang telah dilatih. Kelas dengan probabilitas tertinggi akan dipilih sebagai hasil akhir.

EfficientNet-B4

EfficientNet adalah arsitektur Convolutional Neural Network (CNN) yang dirancang untuk menghasilkan performa klasifikasi citra yang efisien melalui pendekatan *compound scaling* (Tan & Le, 2019). Meskipun pertama kali diperkenalkan pada tahun 2019, studi-studi dalam lima tahun terakhir menunjukkan bahwa EfficientNet masih banyak digunakan dan dievaluasi pada sistem deteksi objek modern sebagai backbone alternatif, termasuk pada integrasi dengan model YOLO, karena kemampuannya dalam menyeimbangkan kualitas ekstraksi fitur dan efisiensi komputasi (Wang et al., 2024; Li et al., 2023).

Dalam hal pengenalan objek, EfficientNet bertugas mengekstrak fitur penting dari gambar, seperti tepi, tekstur, bentuk, atau pola kompleks untuk keperluan pengenalan objek. Proses ini dilakukan melalui lapisan convolutional dan pooling untuk menghasilkan representasi fitur gambar. Setelah fitur diekstraksi, EfficientNet menggunakan *fully connected layers* untuk memetakan fitur tersebut ke dalam kelas-kelas tertentu, seperti "uang kertas Rp50.000" atau "uang kertas Rp100.000." Melalui *compound scaling*, EfficientNet mencapai akurasi tinggi dengan efisiensi skala yang baik, meskipun pada varian EfficientNet-B4 kompleksitas model dan jumlah parameter relatif besar.

EfficientNet dirancang untuk tugas klasifikasi gambar, bukan untuk deteksi objek dimana EfficientNet tidak memiliki kemampuan untuk menentukan lokasi objek dalam gambar juga tidak menggunakan *bounding boxes* untuk mendeteksi keberadaan beberapa objek dalam satu gambar sehingga jika dalam satu gambar terdapat banyak objek, EfficientNet tidak dapat mendeteksi atau membedakan lokasi dan jenis objek secara bersamaan. Karena keterbatasannya tersebut sehingga EfficientNet sering digunakan dalam kombinasi dengan algoritma lain yang memiliki kemampuan deteksi objek seperti YOLO.



Gambar 3. Arsitektur Jaringan EfficientNet

Arsitektur jaringan EfficientNet-B4 seperti ditunjukkan pada Gbr-3 di atas (Pak et al, 2020) secara khusus menunjukkan lapisan-lapisan utama dari EfficientNet-B4, yang terdiri dari blok MBConv (*Mobile Inverted Bottleneck Convolution*), yang bekerja untuk mengekstrak fitur dari gambar secara bertahap. Di sinilah konsep *compound scaling* digunakan untuk mengoptimalkan dimensi jaringan secara bersamaan:

- Depth (kedalaman jaringan) merupakan jumlah lapisan dalam jaringan yang menentukan seberapa dalam model dapat mengekstraksi fitur dari gambar, mulai dari pola sederhana hingga detail kompleks.
- Width (lebar Jaringan) yaitu meningkatkan jumlah saluran (*channels*) dalam setiap lapisan yang mempengaruhi kapasitas model dalam menangkap variasi fitur pada gambar sehingga lebih banyak informasi visual dapat diproses dalam satu lapisan.
- Resolution (resolusi Gambar) merupakan ukuran gambar yang diproses oleh model, yang menentukan seberapa banyak detail yang dapat ditangkap dan dipelajari oleh jaringan.

EfficientNetB4 menggunakan arsitektur khusus yang disebut MBConv dan beberapa blok lainnya untuk mengekstraksi fitur penting dari gambar. Proses ini melibatkan beberapa sub-tahapan:

- Stem Layer: Lapisan awal yang melakukan konvolusi standar untuk mengurangi dimensi gambar.

- MBConv Blocks: Beberapa blok MBConv yang bertugas mengekstraksi fitur dari gambar. Setiap blok MBConv terdiri dari beberapa komponen:
 - a. Depthwise Separable Convolution: Mengurangi jumlah parameter dan operasi komputasi dengan memisahkan konvolusi spasial dan konvolusi channel-wise.
 - b. Squeeze-and-Excitation (SE) Blocks: Menambahkan mekanisme perhatian (attention mechanism) yang membantu model untuk lebih fokus pada fitur-fitur penting.
 - c. Activation Function: Menggunakan fungsi aktivasi Swish yang membantu meningkatkan kinerja model.
 - d. Batch Normalization: Mengurangi variansi internal dalam jaringan untuk mempercepat pelatihan.

Metrik Evaluasi

Penelitian ini menerapkan metrik evaluasi yang lazim digunakan untuk menilai performa model, sehingga kinerja sistem deteksi objek dapat dianalisis secara objektif dan komprehensif. Adapun metrik evaluasi yang digunakan antara lain:

a. Precision

Menggambarkan tingkat ketepatan model dalam memberikan prediksi positif. Metrik ini menunjukkan proporsi prediksi objek yang benar dibandingkan dengan seluruh prediksi objek yang dihasilkan oleh model. Dalam konteks penelitian ini, precision menunjukkan seberapa besar kemungkinan bahwa bounding box yang diprediksi sebagai uang kertas dengan nominal tertentu benar-benar merupakan uang kertas dengan kelas tersebut. Nilai precision yang tinggi menandakan bahwa model jarang menghasilkan deteksi keliru (*false positive*), sehingga hasil deteksi lebih dapat dipercaya.

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

b. Recall

Mengukur kemampuan model dalam menemukan seluruh objek yang sebenarnya ada di dalam gambar. Pada sistem deteksi uang kertas, recall menunjukkan sejauh mana model mampu mendeteksi seluruh uang yang terdapat dalam gambar, termasuk pada kondisi sulit seperti pencahayaan redup, uang terlipat, dan kusam, atau latar belakang yang kompleks. Nilai recall yang rendah mengindikasikan masih adanya objek uang yang terlewatkan oleh sistem (*false negative*).

$$Recall = \frac{TP}{(TP+FN)} \quad (4)$$

c. mean Average Precision (mAP)

mAP mengukur seberapa akurat model dalam menemukan dan mengenali objek yang benar, serta seberapa konsisten model tersebut dalam memberikan hasil yang tepat untuk setiap kelas. Pada penelitian ini digunakan dua jenis mAP, yaitu:

- [mAP@0.5](#): mengukur akurasi deteksi objek dengan toleransi lokasi yang relatif longgar. Suatu prediksi dianggap benar apabila nilai *Intersection over Union (IoU)* antara *bounding box* prediksi dan *ground truth* minimal sebesar 0.5.

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i^{IoU=0.5} \quad (5)$$

dimana N = jumlah kelas (nominal uang kertas)

- [mAP@0.5:0.95](#) merupakan metrik evaluasi yang lebih ketat karena menghitung rata-rata Average Precision pada berbagai ambang IoU, mulai dari 0.5 hingga 0.95 dengan interval 0.05.

$$mAP@0.5:0.95 = \frac{1}{10N} \sum_{t=0.5}^{0.95} \sum_{i=1}^N AP_i^{IoU=t} \quad (6)$$

d. Inference time

Digunakan mengukur waktu yang dibutuhkan model untuk memproses satu gambar dan menghasilkan hasil deteksi. Metrik ini mencerminkan efisiensi komputasi dan menentukan kelayakan model untuk digunakan pada aplikasi *real-time*. Dalam penelitian ini, inference time digunakan untuk membandingkan kecepatan pemrosesan antara YOLOv5 baseline dan YOLOv5 dengan backbone EfficientNet-B4.

$$Inference\ Time = \frac{Total\ waktu\ inferensi}{Jumlah\ citra\ yang\ diproses} \quad (7)$$

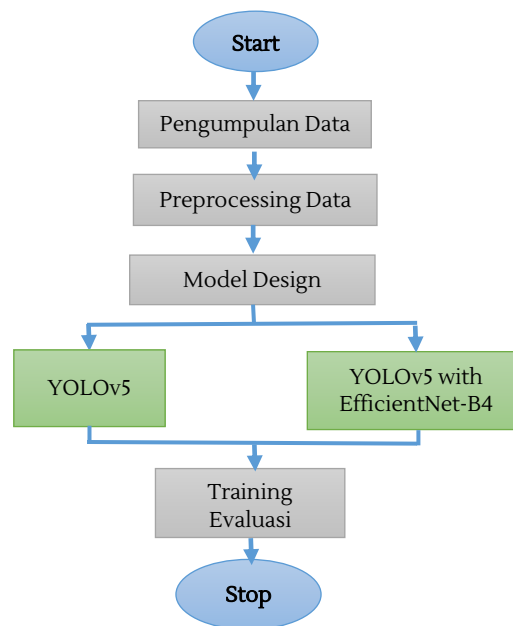
Dalam penelitian ini dilakukan perbandingan antara model deteksi objek, YOLOv5 Baseline dan YOLOv5 dengan backbone EfficientNet-B4 untuk mengetahui seberapa besar peningkatan atau penurunan performa model setelah dilakukan modifikasi arsitektur. Selain itu, pengujian juga melibatkan backbone EfficientNet-B0 dan EfficientNet-B1 sebagai pembanding tambahan guna memberikan gambaran yang lebih komprehensif mengenai pengaruh variasi backbone EfficientNet terhadap kinerja deteksi objek. Berikut ukuran persentase peningkatan (*percentage improvement*) terhadap metrik evaluasi Precision, Recall, mAP@0.5, dan [mAP@0.5:0.95](#).

$$\% Improvement = \frac{Nilai\ model\ baru - Nilai\ baseline}{Nilai\ baseline} \times 100\% \quad (8)$$

METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan metode. Gambar 4 menunjukkan alur penelitian yang terdiri dari empat tahap utama: pengumpulan data, pra-pemrosesan data (*preprocessing*), pemodelan (model design), serta proses pelatihan dan evaluasi model.

Pengumpulan data menjelaskan tentang sumber data dan bagaimana proses pengumpulan data dilakukan. Dataset yang digunakan pada penelitian ini menggunakan dataset gambar uang kertas rupiah emisi 2022 dengan pecahan Rp. 1.000, Rp. 2.000, Rp. 5000, Rp. 10.000, Rp. 20.000, Rp. 50.000, dan Rp. 100.000. Pengumpulan data yang digunakan oleh peneliti adalah sumber data sekunder, karena data yang di peroleh adalah secara tidak langsung atau melalui sumber lain yang sudah tersedia datanya sebelum peneliti melakukan penelitian. Sumber yang digunakan untuk pengumpulan data melalui website <https://roboflow.com>. Pemilihan dataset dilakukan berdasarkan kriteria variasi kondisi fisik uang kertas, seperti tingkat keausan (lusuh, dan terlipat), serta perbedaan pencahayaan dan sudut pengambilan gambar, guna memastikan kemampuan generalisasi model terhadap kondisi nyata.



Gambar 4. Alur Penelitian

Preprocessing menjelaskan tentang pra-pemrosesan data yang bertujuan untuk mempersiapkan data agar lebih mudah digunakan dalam tahap pemodelan. Tahap preprocessing yang dilakukan pada penelitian ini yakni mengubah ukuran gambar (resize) untuk menstandarkan gambar dikarenakan teknik pengambilan gambar yang berbeda-beda. Seluruh gambar pada dataset disesuaikan ukurannya ke resolusi input yang digunakan oleh model YOLOv5 dan EfficientNet-B4 sebelum diproses lebih lanjut, sehingga setiap gambar memiliki dimensi yang seragam dan sesuai dengan kebutuhan arsitektur jaringan. Tujuannya adalah memastikan bahwa input gambar sesuai dengan format yang diperlukan oleh YOLOv5 dan EfficientNet-B4. Kemudian tahap selanjutnya dilakukan Normalisasi dimana nilai piksel gambar (0-255) dinormalisasi ke skala 0-1. Hal ini dilakukan untuk membantu model memproses data lebih stabil dan efisien. Tahap berikutnya dilakukan Augmentasi data untuk meningkatkan variasi gambar sehingga model dapat belajar dari berbagai kondisi. Teknik augmentasi meliputi: Rotasi untuk memutar gambar, Flipping untuk membalik gambar secara horizontal atau vertical, dan Brightness/Contrast untuk mengubah tingkat kecerahan dan kontras gambar. Selanjutnya, dataset dibagi menjadi data latih, data validasi, dan data uji dengan proporsi masing-masing sebesar 70%, 15%, dan 15%. Pembagian dataset ini telah dilakukan sebelumnya melalui platform Roboflow, sehingga pada penelitian ini tidak melakukan proses pembagian dataset kembali untuk menjaga konsistensi dan validitas evaluasi model. Rangkaian proses ini memastikan bahwa gambar uang kertas rupiah siap digunakan untuk melatih model, sehingga model dapat mendeteksi uang kertas dengan lebih akurat di berbagai kondisi.

Pemodelan (*model design*) menjelaskan tentang proses deteksi nilai mata uang menggunakan arsitektur Yolov5 dan EfficientNet-B4. Setelah data diproses, model deteksi nilai mata uang dibangun dengan mengintegrasikan YOLOv5 dengan arsitektur EfficientNet-B4. Model ini dirancang untuk meningkatkan performa YOLOv5 dalam mendeteksi nilai mata uang dengan akurasi deteksi yang lebih baik. Pada penelitian ini akan dibangun dua model deteksi nilai mata uang kertas yaitu model YOLOv5 saja dan

model YOLOv5 yang diintegrasikan dengan arsitektur EfficientNet-B4 sebagai backbone. Hal ini dilakukan untuk membandingkan performa kedua model tersebut sehingga diperoleh pengetahuan bagaimana optimalisasi YOLOv5 dengan arsitektur EfficientNet-B4 sebagai backbone dalam sistem deteksi nilai mata uang.

Tahap training merupakan proses utama dalam penelitian ini yang bertujuan untuk melatih model agar mampu mengenali dan mendeteksi nilai nominal uang kertas secara akurat. Training dilakukan menggunakan dua pendekatan model, yaitu YOLOv5 baseline dan YOLOv5 dengan backbone EfficientNet-B4, menggunakan konfigurasi dataset dan parameter pelatihan yang sama agar perbandingan performa bersifat adil.

Pada YOLOv5 baseline, seluruh komponen jaringan, termasuk backbone, neck, dan head, dilatih secara end-to-end sejak awal. Proses ini memungkinkan model mempelajari fitur visual dan pola objek secara langsung dari dataset uang kertas. Sedangkan pada model YOLOv5 dengan backbone EfficientNet-B4, training dilakukan menggunakan metode bertahap (*freeze* dan *unfreeze*). Konsep *freeze* dan *unfreeze* layer adalah bagian dari strategi *transfer learning* yaitu teknik di mana model yang sudah dilatih pada satu tugas digunakan kembali untuk tugas lain dan disesuaikan. Melakukan *freezing* bagian tertentu dari pretrained model dapat mempengaruhi performa dan efisiensi pelatihan (Dobrzycki et al, 2025).

Pada tahap *freeze*, bobot EfficientNet-B4 dipertahankan tetap dan tidak diperbarui, sehingga proses pembelajaran difokuskan pada bagian neck dan head YOLOv5. Pendekatan ini bertujuan untuk menyesuaikan mekanisme deteksi terhadap fitur-fitur visual yang telah diekstraksi oleh backbone pretrained secara lebih stabil. Selanjutnya, pada tahap *unfreeze*, seluruh bobot jaringan, termasuk backbone EfficientNet-B4, dibuka kembali dan dilatih secara bersamaan. Tahap ini memungkinkan model melakukan penyesuaian lanjutan terhadap karakteristik spesifik dataset uang kertas rupiah, seperti variasi pencahayaan, kondisi fisik uang, dan latar belakang, dengan perubahan bobot yang lebih terkontrol. Dengan pendekatan training ini, model diharapkan mampu mencapai keseimbangan antara stabilitas pembelajaran, akurasi deteksi, dan kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya.

Evaluasi menjelaskan tentang evaluasi kinerja model yang digunakan untuk mengetahui seberapa baik model tersebut bekerja. Setelah pelatihan intensif menggunakan model YOLOv5 dengan CSPDarknet sebagai backbone default YOLOv5 dan YOLOv5 dengan integrasi arsitektur EfficientNet-B4 sebagai backbone. Kedua model dievaluasi menggunakan data uji guna menilai kinerjanya. Evaluasi ini bertujuan untuk melihat kemampuan model dalam mengenali dan mengklasifikasikan data yang belum pernah dipelajari, sehingga mencerminkan penerapannya pada kondisi nyata.

Evaluasi performa kedua model ini dilakukan menggunakan metrik Precision, Recall, mAP@0.5, mAP@0.5:0.95, serta waktu inferensi untuk memberikan gambaran yang lebih menyeluruh mengenai kemampuan model dalam mendeteksi dan mengklasifikasikan objek. Hal ini memungkinkan penilaian kinerja model tidak hanya dari ketepatan dan kelengkapan deteksi, tetapi juga seberapa cepat model bekerja dalam mendeteksi objek. Selanjutnya perhitungan nilai improvement dilakukan untuk menganalisis pengaruh penggunaan backbone EfficientNet-B4 terhadap kinerja YOLOv5 dengan membandingkan performa YOLOv5 baseline dan YOLOv5 setelah penggantian backbone. Melalui perbandingan ini, dapat diketahui apakah penerapan EfficientNet-B4 memberikan peningkatan atau justru penurunan kinerja model dalam mendeteksi dan mengklasifikasikan objek.

HASIL DAN PEMBAHASAN

Konfigurasi dan Skenario Pengujian

Pengembangan arsitektur model, pelatihan dan pengujian diimplementasikan dalam Google Colab Environment dengan menggunakan IPython Widgets digunakan sebagai modul Python utama dalam lingkungan pengembangan. Proses pembuatan dataset, pelabelan, serta pembagian data (*split balancing*) dilakukan menggunakan Roboflow, kemudian dataset diunduh dan disimpan pada Local SSD (*Solid State Drive*) sebagai media penyimpanan utama selama proses pelatihan dan evaluasi model. Penggunaan Local SSD dipilih karena memiliki kecepatan akses data yang lebih tinggi dan tidak bergantung pada koneksi jaringan, sehingga mampu mengurangi hambatan *input/output* yang sering terjadi saat menggunakan Google Drive. Dengan demikian, proses pelatihan menjadi lebih stabil, efisien, dan konsisten, terutama pada pelatihan model deteksi objek yang membutuhkan akses data secara intensif.

Dataset yang digunakan dalam penelitian ini terdiri dari 2.284 gambar uang kertas rupiah yang diambil dari berbagai sudut dan kondisi pencahayaan. Dataset ini telah dibagi menjadi menjadi tiga bagian yaitu data latih (train) sebanyak 1.599 gambar, data validasi (validation) sebanyak 343 gambar, dan data uji (test) sebanyak 342 gambar. Pembagian dataset ini bertujuan untuk memastikan proses pelatihan model berjalan optimal, sekaligus memungkinkan evaluasi performa model secara objektif menggunakan data yang tidak pernah dilihat sebelumnya.

Sebelum seluruh proses pelatihan dan pengujian dilakukan, terlebih dahulu dilakukan pengaturan terhadap konfigurasi parameter dan strategi pelatihan model untuk memastikan bahwa perbandingan kinerja antara YOLOv5 baseline dan YOLOv5 dengan backbone EfficientNet-B4 dilakukan secara objektif dan terkontrol. Rincian pengaturan parameter dan strategi pelatihan model yang digunakan dapat dilihat pada Tabel 1 berikut.

Tabel 1. Parameter dan Strategi Pelatihan

Model	Parameter	Value
YOLOv5 Baseline (CSPDarknet Backbone)	Pretrained	YOLOv5s pretrained COCO
	Image Size	640 × 640
	Batch Size	16
	Epoch	30
	Optimizer	SGD
	Learning Rate	0.01
	Device	GPU (CUDA)
YOLOv5 + EfficientNet-B4 (Stage A – Freeze Backbone)	Backbone	Freeze
	Image Size	640 × 640
	Batch Size	4
	Epoch	30
	Optimizer	AdamW
	Learning Rate	0.001
	Device	GPU (CUDA)
YOLOv5 + EfficientNet-B4 (Stage B – Unfreeze Backbone)	Backbone	Unfreeze
	Image Size	640 × 640
	Batch Size	4

	Epoch	50
	Optimizer	AdamW
	Learning Rate	0.0005
	Bobot Awal	Hasil Stage A
	Device	GPU (CUDA)

Pengujian dilakukan dalam tiga tahap, masing-masing dengan parameter atau kondisi tertentu:

1. Deteksi nilai mata uang dengan model YOLOv5 Default (CSPDarknet) sebagai baseline backbone.
2. Deteksi nilai mata uang dengan model YOLOv5 dengan arsitektur EfficientNet-B4 sebagai backbone.
3. Deteksi nilai mata uang dengan model YOLOv5 dengan arsitektur EfficientNet-B0 dan B1 sebagai backbone

Hasil training model

Model YOLOv5 dengan backbone EfficientNet-B4 dilatih menggunakan pendekatan dua tahap (*two-stage training*), yaitu:

- Stage A: Backbone EfficientNet-B4 dibekukan (*freeze*) selama 30 epoch, menggunakan optimizer AdamW dan cosine learning rate scheduler. Tahap ini bertujuan untuk menyesuaikan head deteksi YOLO terhadap karakteristik fitur dari EfficientNet tanpa mengubah bobot backbone secara signifikan.
- Stage B: Seluruh layer dibuka (*unfreeze*) dan dilatih ulang selama 50 epoch menggunakan optimizer AdamW dengan cosine learning rate. Bobot awal pada tahap ini diambil dari hasil Stage A. Strategi ini bertujuan untuk menyempurnakan proses pembelajaran fitur secara menyeluruh.

Sedangkan Ukuran batch pada kedua tahap ditetapkan sebesar 4, menyesuaikan keterbatasan memori GPU akibat penggunaan backbone yang lebih kompleks. Adapun hasil training kedua model ditunjukkan Gambar 5 dan Gambar 6 berikut:

```

30 epochs completed in 0.332 hours.
Optimizer stripped from runs/train/baseline_yolov5s_rupiah/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/baseline_yolov5s_rupiah/weights/best.pt, 14.5MB

Validating runs/train/baseline_yolov5s_rupiah/weights/best.pt...
Fusing layers...
YOLOv5s_rupiah summary: 157 layers, 7029004 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95	100% 11/11 [00:06<00:00, 1.81it/s]
all	343	720	0.651	0.901	0.848	0.716	
1K	343	77	0.726	0.961	0.883	0.737	
2K	343	59	0.916	0.927	0.97	0.782	
5K	343	106	0.582	0.962	0.872	0.768	
10K	343	110	0.547	0.682	0.752	0.64	
20K	343	96	0.604	0.939	0.813	0.702	
50K	343	138	0.499	0.891	0.794	0.668	
100K	343	134	0.683	0.948	0.855	0.713	

```

Results saved to runs/train/baseline_yolov5s_rupiah

```

Gambar 5. Hasil Training Yolov5 Baseline

Secara umum, hasil training YOLOv5 baseline mampu mendeteksi seluruh nominal uang kertas dengan performa yang cukup baik. Nominal Rp2.000 dan Rp1.000 menunjukkan hasil terbaik, sedangkan nominal Rp10.000 menjadi kelas yang paling menantang bagi model.

```

50 epochs completed in 3.802 hours.
Optimizer stripped from runs/train/effb4_stageB_lr0005/weights/last.pt, 42.1MB
Optimizer stripped from runs/train/effb4_stageB_lr0005/weights/best.pt, 42.1MB

Validating runs/train/effb4_stageB_lr0005/weights/best.pt...
Model summary: 604 layers, 20709556 parameters, 0 gradients

```

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95	100% 43/43 [00:09<00:00, 4.33it/s]
all	343	720	0.669	0.873	0.841	0.733	
1K	343	77	0.742	0.971	0.872	0.74	
2K	343	59	0.901	0.924	0.967	0.819	
5K	343	106	0.58	0.934	0.853	0.761	
10K	343	110	0.642	0.6	0.762	0.679	
20K	343	96	0.613	0.875	0.809	0.721	
50K	343	138	0.533	0.9	0.788	0.678	
100K	343	134	0.669	0.905	0.836	0.731	

```

Results saved to runs/train/effb4_stageB_lr0005

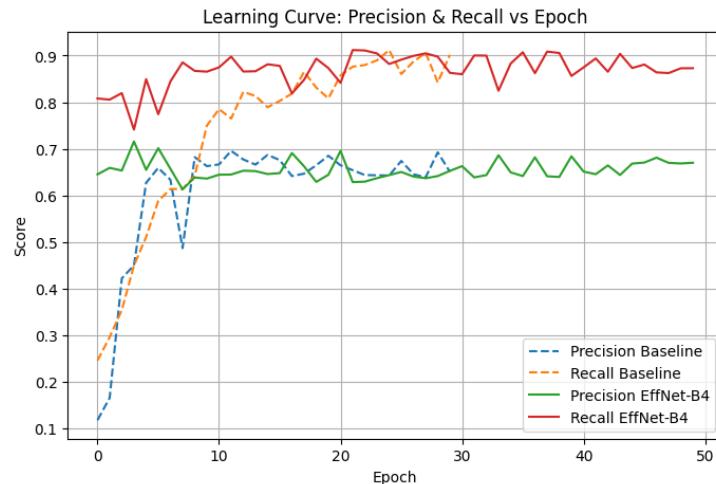
```

Gambar 6. Hasil Training Yolov5 dengan Backbone EfficientNet-B4

Model YOLOv5 dengan backbone EfficientNet-B4 dilatih selama 50 epoch dan menghasilkan performa yang relatif stabil pada seluruh kelas nominal uang rupiah. Arsitektur EfficientNet-B4 memiliki jumlah parameter yang jauh lebih besar dibandingkan YOLOv5 baseline, sehingga diharapkan mampu mengekstraksi fitur visual yang lebih kaya dan detail. Hal ini tercermin dari peningkatan pada beberapa metrik evaluasi, khususnya yang berkaitan dengan kualitas lokalisasi objek. Model mencapai nilai precision sebesar 66.9, recall sebesar 87.3, mAP@0.5 sebesar 84.1, dan mAP@0.5:0.95 sebesar 73.3. Nilai recall yang relatif tinggi menunjukkan bahwa model cukup andal dalam menemukan objek uang kertas, sementara nilai mAP@0.5:0.95 yang cukup baik mengindikasikan kualitas lokalisasi bounding box yang lebih konsisten pada berbagai ambang IoU.

Hasil training menunjukkan bahwa model YOLOv5 dengan backbone EfficientNet-B4 mampu mendeteksi seluruh nominal uang rupiah dengan tingkat keberhasilan yang baik. Nominal Rp2.000 dan Rp1.000 menunjukkan performa paling seimbang dan konsisten, sedangkan nominal Rp10.000 menjadi kelas yang sulit dikenali model. Beberapa kelas seperti Rp5.000 dan Rp50.000 memiliki recall tinggi namun precision rendah, yang menunjukkan bahwa model efektif dalam menemukan objek tetapi masih perlu perbaikan dalam ketepatan klasifikasi.

Analisis dinamika proses pelatihan model tidak hanya dilakukan pada hasil evaluasi akhir, tetapi juga pada perkembangan metrik kinerja selama pelatihan. Learning curve ditunjukkan Gambar 7, digunakan untuk menggambarkan perubahan nilai precision dan recall terhadap jumlah epoch sehingga memungkinkan pengamatan terhadap pola pembelajaran, stabilitas prediksi, dan proses konvergensi model. Pada penelitian ini, learning curve dianalisis untuk membandingkan perilaku pembelajaran antara model YOLOv5 baseline dan model YOLOv5 dengan backbone EfficientNet-B4 sebagai dasar dalam menjelaskan perbedaan karakteristik kinerja kedua model.

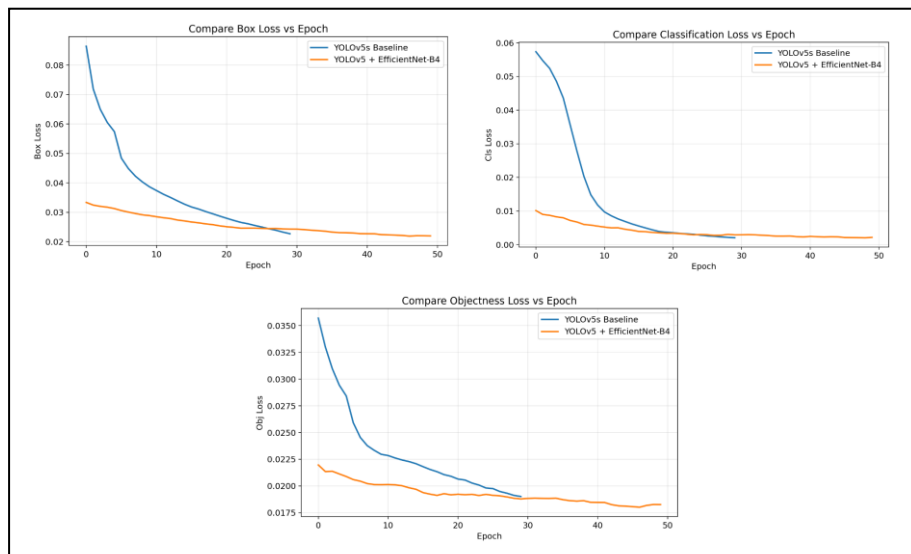


Gambar 7. Learning Kurva Precision dan Recall

Analisis *learning curve* menunjukkan bahwa pada fase awal pelatihan (epoch 0–10), model YOLOv5 baseline mengalami peningkatan recall yang cepat, namun disertai fluktuasi precision yang cukup besar, mengindikasikan tingginya tingkat prediksi positif yang kurang akurat. Sebaliknya, model dengan backbone EfficientNet-B4 memperlihatkan precision yang lebih stabil sejak awal, meskipun peningkatan recall berlangsung lebih bertahap, yang mencerminkan proses pembelajaran yang lebih terkontrol. Pada fase konvergensi menengah (epoch 10–30), kedua model mulai menunjukkan kestabilan, namun EfficientNet-B4 mempertahankan konsistensi precision yang lebih baik seiring peningkatan recall yang stabil. Pada fase akhir pelatihan (epoch >30), kedua model telah mencapai kondisi konvergen, di mana model baseline mempertahankan recall tinggi dengan precision yang relatif lebih rendah, sementara EfficientNet-B4 menunjukkan kombinasi precision dan recall yang lebih seimbang dengan fluktuasi minimal, menandakan proses pembelajaran yang lebih konsisten secara keseluruhan.

Dari *learning curve* ini dapat disimpulkan bahwa model YOLOv5 baseline cenderung memprioritaskan kemampuan mendeteksi objek sejak awal pelatihan, yang tercermin dari peningkatan recall yang cepat namun disertai fluktuasi precision. Sebaliknya, model YOLOv5 dengan backbone EfficientNet-B4 menunjukkan proses pembelajaran yang lebih stabil dan terkontrol, ditandai dengan precision yang konsisten serta pola konvergensi yang lebih halus selama pelatihan. Hal ini sejalan dengan strategi *freeze* dan *unfreeze layer* dalam *transfer learning*. Dobrzycki et al (2025) menyatakan bahwa penerapan *freezing* pada lapisan tertentu pada model *pretrained* dapat meningkatkan stabilitas pelatihan dan mempercepat konvergensi dengan mengurangi fluktuasi gradien pada tahap awal pembelajaran. Selain itu, pendekatan ini juga mampu menurunkan penggunaan memori unit pemrosesan grafis (GPU) hingga 28% dibandingkan dengan metode *full fine-tuning*.

Dalam melengkapi analisis hasil pelatihan, dilakukan pengamatan terhadap perilaku kurva loss selama proses training seperti ditunjukkan pada Gambar 8. Kurva box loss, objectness loss, dan classification loss disajikan untuk menggambarkan dinamika pembelajaran model dalam mengoptimalkan lokalisasi objek, pendeteksian keberadaan objek, dan klasifikasi kelas seiring bertambahnya epoch. Analisis kurva loss ini memberikan konteks tambahan dalam menafsirkan kinerja model, khususnya terkait kestabilan proses pelatihan dan kecenderungan konvergensi yang dicapai oleh masing-masing arsitektur backbone.



Gambar 8. Kurva Loss (box/obj/cls)

Kurva Box Loss memperlihatkan bagaimana kemampuan model dalam melokalisasi bounding box (posisi dan ukuran kotak deteksi) berkembang sepanjang pelatihan. Kurva Box Loss pada YOLOv5 baseline dimulai cukup tinggi (sekitar $>0,08$) lalu turun tajam pada epoch awal. Penurunan curam di 5–10 epoch pertama menandakan fase adaptasi cepat dimana model sedang belajar pola global objek dan menyesuaikan regresi bounding box dari kondisi awal yang masih kasar. Setelah itu laju penurunan melambat dan kurva cenderung mendekati plateau hingga sekitar epoch 25–30, yang menunjukkan model mulai memasuki fase konvergensi lokalisasi, perbaikan masih terjadi, tetapi incremental (semakin kecil). Pola ini umum terjadi pada backbone yang lebih ringan, konvergensi cepat, tetapi setelah titik tertentu perbaikan box menjadi terbatas karena kapasitas fitur dan resolusi representasi yang tersedia.

Berbeda dengan baseline, model YOLOv5 dengan backbone EfficientNet-B4 memulai box loss jauh lebih rendah (sekitar 0,033) dan menurun secara lebih halus namun konsisten hingga akhir training (mendekati 0,022). Ada dua makna teknis penting di sini. Pertama, nilai awal yang lebih rendah mengindikasikan bahwa sejak awal pelatihan model sudah memiliki fondasi fitur yang lebih kuat untuk memandu regresi box, terutama bila backbone memanfaatkan bobot pretrained dan strategi freeze–unfreeze, sehingga head deteksi tidak berjuang dari nol untuk menemukan representasi yang relevan. Kedua, penurunan yang stabil hingga epoch 50 menunjukkan proses optimasi lokalisasi masih berlangsung dan belum benar-benar jenuh pada epoch pertengahan. Hal ini selaras dengan karakter backbone yang lebih kompleks, yang cenderung membutuhkan waktu lebih panjang untuk memaksimalkan pembelajaran fitur.

Kurva Classification Loss (cls loss) menggambarkan kemampuan model dalam membedakan dan mengklasifikasikan kelas objek, dalam penelitian ini berupa nominal uang kertas rupiah. Semakin kecil nilai cls loss, semakin baik model dalam mempelajari perbedaan karakteristik visual antar kelas dan mengurangi kesalahan prediksi kelas. Pada model YOLOv5 baseline, nilai cls loss pada awal pelatihan relatif tinggi dan mengalami penurunan yang cukup tajam pada sekitar 10 epoch pertama. Pola ini menunjukkan bahwa model baseline membutuhkan fase adaptasi awal yang cukup signifikan untuk

mempelajari fitur diskriminatif antar kelas. Setelah epoch tersebut, penurunan cls loss melambat dan cenderung stabil hingga mendekati akhir pelatihan, yang menandakan bahwa kemampuan klasifikasi model telah mencapai kondisi konvergensi lebih awal. Sebaliknya, model YOLOv5 dengan backbone EfficientNet-B4 menunjukkan nilai cls loss yang jauh lebih rendah sejak epoch awal dan menurun secara bertahap serta stabil sepanjang proses pelatihan hingga epoch ke-50. Kondisi ini mengindikasikan bahwa EfficientNet-B4 mampu menyediakan representasi fitur yang lebih kaya dan informatif untuk mendukung proses klasifikasi kelas sejak tahap awal training. Penurunan cls loss yang konsisten juga menunjukkan bahwa model masih terus melakukan perbaikan dalam membedakan kelas nominal, meskipun laju perbaikannya relatif lebih lambat dibandingkan fase awal pada model baseline.

Kurva Objectness Loss (obj loss) merepresentasikan kemampuan model dalam memprediksi keberadaan objek pada setiap kandidat bounding box, yaitu membedakan antara area yang benar-benar mengandung objek dan area latar belakang. Nilai objectness loss yang semakin kecil menunjukkan bahwa model semakin mampu mengidentifikasi lokasi objek secara tepat dan mengurangi kesalahan deteksi objek palsu (*false positive*). Pada model YOLOv5 baseline, nilai objectness loss pada epoch awal relatif tinggi dan mengalami penurunan yang cukup tajam pada beberapa epoch pertama. Pola ini menunjukkan bahwa model baseline membutuhkan fase adaptasi awal yang lebih besar untuk mempelajari distribusi objek dan latar belakang pada dataset. Setelah memasuki sekitar epoch ke-10, penurunan obj loss mulai melambat dan cenderung stabil hingga akhir pelatihan, yang mengindikasikan bahwa kemampuan model dalam mengenali keberadaan objek telah mendekati kondisi konvergensi.

Sementara itu, model YOLOv5 dengan backbone EfficientNet-B4 menunjukkan nilai objectness loss yang lebih rendah sejak awal pelatihan dan menurun secara lebih halus serta konsisten hingga epoch terakhir. Kondisi ini mengindikasikan bahwa backbone EfficientNet-B4 mampu menyediakan representasi fitur yang lebih informatif untuk membedakan objek dan latar belakang, sehingga proses pembelajaran objectness dapat berlangsung lebih stabil. Penurunan obj loss yang bertahap juga menunjukkan bahwa model terus melakukan penyempurnaan dalam memprediksi keberadaan objek, meskipun dengan laju yang lebih lambat dibandingkan penurunan tajam pada model baseline. Jika dibandingkan secara langsung, nilai objectness loss pada EfficientNet-B4 berada di bawah nilai objectness loss YOLOv5 baseline pada hampir seluruh rentang epoch. Hal ini menunjukkan bahwa model dengan backbone yang lebih kompleks cenderung lebih efektif dalam mengurangi kesalahan deteksi objek palsu.

Hasil evaluasi model

Hasil training digunakan untuk menentukan apakah model telah mempelajari pola data secara memadai sebelum dilakukan evaluasi lebih lanjut pada data uji. Evaluasi ini dilakukan guna menilai kinerja masing-masing model dalam mendeteksi dan mengklasifikasikan nilai nominal uang kertas pada data uji yang belum pernah digunakan selama proses pelatihan. Hasil evaluasi kedua model ditunjukkan Gambar 9 dan Gambar 10 berikut:

```

/content/yolov5
val: data=/content/dataset_uang_OK/banknote.yaml, weights=['/content/yolov5/runs/train/baseline_yolov5s_rupiah/weights/best.pt'], bat
YOLOv5 v7.0-453-geed9bc19 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

Fusing layers...
YOLOv5s_rupiah summary: 157 layers, 7029004 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning /content/dataset_uang_OK/labels/test... 342 images, 0 backgrounds, 0 corrupt: 100% 342/342 [00:00<00:00, 1690.07it/s]
test: New cache created: /content/dataset_uang_OK/labels/test.cache

```

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	11/11 [00:07<00:00, 1.54it/s]
all	342	742	0.664	0.907	0.837	0.712	
1K	342	54	0.737	0.944	0.828	0.697	
2K	342	28	0.858	0.929	0.969	0.817	
5K	342	138	0.594	0.954	0.82	0.715	
10K	342	135	0.592	0.719	0.771	0.653	
20K	342	143	0.668	0.942	0.839	0.711	
50K	342	128	0.568	0.945	0.834	0.71	
100K	342	116	0.629	0.914	0.801	0.677	

```

Speed: 0.2ms pre-process, 7.0ms inference, 2.9ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/exp

```

Gambar 8. Hasil Evaluasi Yolov5 Baseline

Berdasarkan hasil evaluasi pada data uji, model YOLOv5 baseline menunjukkan kinerja yang baik dan stabil dalam mendeteksi uang kertas rupiah. Nilai recall yang tinggi mengindikasikan bahwa model memiliki kemampuan yang sangat baik dalam menemukan objek uang kertas, sehingga risiko objek tidak terdeteksi (*false negative*) relatif kecil. Nilai precision yang berada pada tingkat menengah menunjukkan bahwa sebagian prediksi masih mengandung kesalahan klasifikasi, namun tidak bersifat dominan. Nilai $mAP@0.5$ yang tinggi menunjukkan bahwa model mampu mendeteksi objek dengan baik pada ambang IoU standar, sedangkan nilai $mAP@0.5:0.95$ yang masih cukup tinggi menunjukkan bahwa model juga memiliki ketelitian lokalisasi bounding box yang memadai pada ambang IoU yang lebih ketat. Dari sisi efisiensi komputasi, waktu inferensi rata-rata sebesar 7 ms per gambar menunjukkan bahwa YOLOv5 baseline sangat efisien dan cocok untuk kebutuhan deteksi real-time.

```

AttributeError: 'FreeTypeFont' object has no attribute 'getsize'

```

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100%	43/43 [00:15<00:00, 2.71it/s]
all	342	742	0.648	0.886	0.828	0.728	
1K	342	54	0.711	0.926	0.799	0.706	
2K	342	28	0.815	0.964	0.967	0.848	
5K	342	138	0.622	0.949	0.825	0.73	
10K	342	135	0.594	0.667	0.768	0.659	
20K	342	143	0.635	0.888	0.815	0.707	
50K	342	128	0.561	0.93	0.819	0.728	
100K	342	116	0.6	0.879	0.806	0.721	

```

Speed: 0.3ms pre-process, 37.8ms inference, 1.2ms NMS per image at shape (8, 3, 640, 640)
Results saved to scripts/runs/val/exp

```

Gambar 9. Hasil Evaluasi Yolov5 dengan Backbone EfficientNet-B4

Hasil evaluasi pada data uji menunjukkan bahwa model YOLOv5 dengan backbone EfficientNet-B4 menunjukkan performa deteksi yang baik dengan karakteristik utama berupa kemampuan deteksi objek yang tinggi dan ketelitian lokalisasi yang relatif stabil. Secara keseluruhan, model memperoleh precision sebesar 64.8, recall sebesar 88.6, $mAP@0.5$ sebesar 82.8, dan $mAP@0.5:0.95$ sebesar 72.8.

Nilai recall yang tinggi mengindikasikan bahwa model mampu mendeteksi sebagian besar objek uang kertas, sehingga risiko objek terlewat (*false negative*) relatif rendah. Precision yang berada pada tingkat menengah menunjukkan bahwa masih terdapat sejumlah prediksi yang kurang tepat, namun tidak bersifat dominan. Nilai $mAP@0.5$ yang cukup tinggi menandakan performa deteksi yang baik pada ambang IoU standar, sementara peningkatan nilai $mAP@0.5:0.95$ menunjukkan bahwa penggunaan EfficientNet-B4 memberikan kontribusi positif terhadap ketelitian lokalisasi bounding box pada ambang IoU yang lebih ketat. Dari sisi efisiensi komputasi, waktu inferensi rata-rata

sebesar 37,8 ms per gambar menunjukkan bahwa model memiliki beban komputasi yang lebih tinggi dibandingkan YOLOv5 baseline. Hal ini merupakan konsekuensi dari kompleksitas arsitektur EfficientNet-B4 yang memiliki jumlah parameter lebih besar.

Secara keseluruhan, YOLOv5 dengan backbone EfficientNet-B4 menunjukkan kinerja deteksi yang baik dengan peningkatan pada aspek ketelitian lokalisasi bounding box, sebagaimana tercermin dari nilai $mAP@0.5:0.95$ yang relatif tinggi. Nominal Rp2.000 menjadi kelas dengan performa terbaik, sedangkan nominal Rp10.000 tetap menjadi kelas yang paling sulit dikenali dibandingkan nominal lainnya karena memiliki kompleksitas visual yang lebih tinggi dibanding nominal lainnya. Warna dominan pada uang Rp10.000 relatif kurang kontras sehingga tidak menonjol secara visual terhadap latar belakang. Selain itu, dataset yang digunakan pada citra uang Rp10.000 memperlihatkan variasi kondisi yang lebih beragam seperti pencahayaan yang tidak merata, keberadaan bayangan, serta kondisi fisik uang yang terlipat. Kondisi-kondisi tersebut menyebabkan proses ekstraksi fitur visual oleh model menjadi kurang optimal. Meskipun demikian, nilai precision, recall, dan $mAP@0.5:0.95$ yang masih berada di atas 60% menunjukkan bahwa model tetap mampu melakukan deteksi dan lokalisasi nominal Rp10.000 dengan tingkat kinerja yang tergolong baik

Tingginya nilai recall pada sebagian besar kelas menunjukkan bahwa model efektif dalam menemukan objek, namun variasi nilai precision menunjukkan bahwa masih terdapat tantangan pada ketepatan klasifikasi antar nominal. Hasil ini menegaskan bahwa penggunaan backbone yang lebih kompleks mampu meningkatkan kualitas representasi fitur, tetapi dengan konsekuensi peningkatan beban komputasi.

Tabel 2 menunjukkan hasil perbandingan kinerja antara model YOLOv5 baseline dan YOLOv5 dengan backbone EfficientNet-B4 pada data uji. Dengan menggunakan rumus percentage improvement (*% Improvement*), penelitian ini bertujuan untuk mengetahui besarnya perubahan kinerja model setelah dilakukan penggantian backbone, baik berupa peningkatan maupun penurunan performa, pada metrik evaluasi Precision, Recall, $mAP@0.5$, dan $mAP@0.5:0.95$ jika dibandingkan dengan model YOLOv5 baseline seperti ditunjukkan pada Tabel 2 berikut:

Tabel 2. Model Evaluation Report

Metrik	YOLOv5 baseline	YOLOv5 + EfficientNet-B4	% Improvement
Precision	66.4	64.8	-2.409
Recall	90.7	88.6	-2.315
$mAP@0.5$	83.7	82.8	-1.075
$mAP@0.5:0.95$	71.2	72.8	2.247
Inference Time	≈ 7.0 ms	≈ 37.8 ms	440

Berdasarkan hasil evaluasi pada data uji, perbedaan kinerja antara model YOLOv5 baseline dan YOLOv5 dengan backbone EfficientNet-B4 bersifat marginal pada sebagian besar metrik evaluasi. Pada metrik precision menunjukkan bahwa model YOLOv5 baseline cenderung lebih akurat dalam memprediksi kelas objek dan menghasilkan lebih sedikit kesalahan prediksi positif dibandingkan model dengan backbone EfficientNet-B4. Pada metrik recall, YOLOv5 baseline juga menunjukkan performa yang lebih baik, hal ini mengindikasikan bahwa model baseline lebih konsisten dalam mendeteksi hampir seluruh objek uang kertas pada citra uji, sementara model dengan backbone EfficientNet-B4 sedikit lebih sering melewatkan objek tertentu. Selain itu evaluasi pada $mAP@0.5$

memperlihatkan bahwa YOLOv5 baseline masih unggul tipis dibandingkan model YOLOv5 dengan backbone EfficientNet-B4. Hal ini menunjukkan bahwa backbone default YOLOv5 sudah cukup efektif pada ambang Intersection over Union (IoU) standar. Namun demikian, pada metrik $mAP@0.5:0.95$, model dengan backbone EfficientNet-B4 justru menunjukkan peningkatan performa. Peningkatan ini mengindikasikan bahwa EfficientNet-B4 mampu menghasilkan lokalisasi bounding box yang lebih presisi ketika dievaluasi pada ambang IoU yang lebih ketat.

Temuan penelitian ini sejalan dengan hasil penelitian Nurmaini dan Pratama (2023) yang menunjukkan bahwa penggunaan backbone EfficientNet-B0 mampu menghasilkan nilai Average Precision tertinggi pada ambang IoU yang sangat ketat (IoU 0,95), meskipun backbone default YOLOv4 tetap menunjukkan performa yang stabil pada metrik evaluasi umum. Temuan ini mengindikasikan bahwa backbone EfficientNet memiliki keunggulan dalam meningkatkan ketelitian lokalisasi bounding box ketika dievaluasi pada standar presisi yang lebih tinggi.

Dari sisi proses pelatihan, kinerja model juga dipengaruhi oleh keterbatasan jumlah data pelatihan dan durasi training. EfficientNet-B4 merupakan backbone dengan tingkat kompleksitas yang tinggi dan jumlah parameter yang besar, sehingga membutuhkan dataset yang lebih banyak serta jumlah epoch pelatihan yang memadai agar proses pembelajaran fitur dapat mencapai kondisi konvergensi yang optimal. Dalam penelitian ini, jumlah epoch pelatihan dibatasi hingga 50 epoch sebagai konsekuensi dari keterbatasan sumber daya GPU. Kondisi tersebut berpotensi menyebabkan model belum sepenuhnya memanfaatkan kapasitas representasi fitur yang dimiliki oleh EfficientNet-B4.

Peningkatan kompleksitas backbone berdampak langsung pada efisiensi komputasi. Waktu inferensi model YOLOv5 dengan backbone EfficientNet-B4 meningkat secara signifikan dibandingkan model baseline, yaitu dari sekitar 7,0 ms menjadi 37,8 ms per gambar. Hal ini menunjukkan bahwa penggunaan backbone yang lebih kompleks memberikan beban komputasi yang lebih besar dan menurunkan kelayakan model untuk aplikasi real-time. Oleh karena itu, pemilihan backbone perlu disesuaikan dengan kebutuhan sistem, terutama terkait keseimbangan antara ketelitian lokalisasi dan kecepatan pemrosesan.

Penelitian ini bertujuan untuk mengevaluasi secara eksperimental penggunaan arsitektur EfficientNet-B4 sebagai backbone pengganti backbone default YOLOv5 (CSPDarknet) dalam sistem deteksi nilai mata uang rupiah. Evaluasi dilakukan untuk meninjau pengaruh penggantian backbone terhadap kinerja deteksi objek, yang meliputi kemampuan deteksi, ketelitian lokalisasi bounding box, serta efisiensi komputasi model. Berdasarkan hasil evaluasi penggunaan backbone EfficientNet-B4, penelitian ini selanjutnya tidak hanya berfokus pada backbone dengan kompleksitas tinggi, tetapi juga melibatkan EfficientNet-B0 dan EfficientNet-B1 sebagai pembanding dengan kompleksitas yang lebih ringan. Kedua backbone tersebut memiliki jumlah parameter yang lebih sedikit dan kebutuhan komputasi yang lebih rendah, sehingga memungkinkan proses pelatihan yang lebih efisien pada kondisi keterbatasan data dan sumber daya komputasi. Pendekatan ini bertujuan untuk memperoleh analisis yang lebih komprehensif mengenai hubungan antara tingkat kompleksitas backbone dan performa deteksi YOLOv5, sehingga dapat memberikan gambaran yang lebih objektif dalam menentukan konfigurasi backbone yang sesuai untuk kebutuhan aplikasi tertentu. Berikut adalah hasil evaluasi YOLOv5 dengan Backbone EfficientNet-B0 dan EfficientNet-B1.

```

AttributeError: 'FreeTypeFont' object has no attribute 'getsize'
Class      Images      Labels      P      R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:08<00:00, 5.15it/s]
all        342          742        0.639  0.892  0.833    0.683
1K         342          54         0.757  0.963  0.843    0.659
2K         342          28         0.789  0.938  0.944    0.735
5K         342          138        0.585  0.95   0.821    0.689
10K        342          135        0.551  0.718  0.776    0.646
20K        342          143        0.634  0.916  0.826    0.684
50K        342          128        0.565  0.913  0.832    0.703
100K       342          116        0.595  0.845  0.791    0.667
Speed: 0.4ms pre-process, 15.3ms inference, 1.3ms NMS per image at shape (8, 3, 640, 640)
Results saved to scripts/runs/val/exp

```

Gambar 10. Hasil Evaluasi Yolov5 dengan Backbone EfficientNet-B0

```

AttributeError: 'FreeTypeFont' object has no attribute 'getsize'
Class      Images      Labels      P      R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:09<00:00, 4.32it/s]
all        342          742        0.715  0.809  0.843    0.716
1K         342          54         0.738  0.907  0.872    0.736
2K         342          28         0.925  0.857  0.967    0.8
5K         342          138        0.65   0.92   0.827    0.71
10K        342          135        0.776  0.556  0.776    0.659
20K        342          143        0.676  0.79   0.829    0.699
50K        342          128        0.604  0.799  0.822    0.706
100K       342          116        0.637  0.836  0.806    0.705
Speed: 0.3ms pre-process, 20.5ms inference, 1.3ms NMS per image at shape (8, 3, 640, 640)
Results saved to scripts/runs/val/exp

```

Gambar 11. Hasil Evaluasi Yolov5 dengan Backbone EfficientNet-B1

Guna memperoleh gambaran yang lebih komprehensif mengenai pengaruh tingkat kompleksitas backbone terhadap kinerja YOLOv5, dilakukan perbandingan antara beberapa varian EfficientNet dengan kapasitas yang berbeda, yaitu EfficientNet-B0, EfficientNet-B1, dan EfficientNet-B4. Ketiga backbone tersebut dipilih untuk merepresentasikan backbone ringan, menengah, dan kompleks, sehingga memungkinkan analisis yang lebih menyeluruh terhadap hubungan antara kapasitas representasi fitur, kinerja deteksi, dan efisiensi komputasi. Hasil evaluasi masing-masing konfigurasi backbone pada data uji disajikan secara ringkas dalam Tabel 3 yang memuat perbandingan metrik evaluasi utama serta karakteristik kinerja dari setiap model.

Tabel 3. Kinerja YOLOv5 dengan Backbone EfficientNet-B0, B1, dan B4

Metrik	EfficientNet-B0	EfficientNet-B1	EfficientNet-B4
Precision	63.9	71.5	64.8
Recall	89.2	80.9	88.6
mAP@0.5	83.3	84.3	82.8
mAP@0.5:0.95	68.3	71.6	72.8
Inference Time	≈ 15.3 ms	≈ 20.5 ms	≈ 37.8 ms

Berdasarkan hasil evaluasi pada data uji, penggunaan backbone EfficientNet dengan tingkat kompleksitas yang berbeda menghasilkan karakteristik kinerja yang beragam pada model YOLOv5. Backbone EfficientNet-B0, sebagai varian paling ringan, menunjukkan keunggulan utama pada aspek efisiensi komputasi dengan waktu inferensi yang relatif cepat. Meskipun demikian, keterbatasan kapasitas representasi fitur menyebabkan nilai mAP@0.5:0.95 yang dihasilkan masih lebih rendah dibandingkan backbone lainnya yang lebih kompleks.

Backbone EfficientNet-B1 menunjukkan peningkatan kinerja yang paling seimbang. Nilai precision dan mAP@0.5 yang tertinggi di antara ketiga backbone mengindikasikan bahwa EfficientNet-B1 mampu menghasilkan prediksi kelas yang lebih akurat tanpa meningkatkan beban komputasi secara signifikan. Dengan waktu inferensi yang masih

relatif moderat, backbone ini mampu memanfaatkan kapasitas representasi fitur secara efektif dalam kondisi keterbatasan jumlah data, epoch pelatihan, dan sumber daya GPU.

Sementara itu, EfficientNet-B4 sebagai backbone dengan kompleksitas tertinggi menunjukkan keunggulan pada metrik $mAP@0.5:0.95$, yang menandakan peningkatan ketelitian lokalisasi bounding box. Namun, peningkatan tersebut tidak diikuti oleh peningkatan signifikan pada $mAP@0.5$, precision, maupun recall. Selain itu, waktu inferensi yang meningkat secara signifikan menunjukkan bahwa backbone ini memberikan beban komputasi yang besar, sehingga kurang sesuai untuk aplikasi yang memerlukan pemrosesan *real-time*. Temuan ini sejalan dengan studi oleh Tan dan Le (2021), yang menunjukkan bahwa peningkatan skala EfficientNet dari varian yang lebih kecil ke varian yang lebih besar secara konsisten diikuti oleh peningkatan jumlah parameter, FLOPs, dan waktu inferensi, meskipun peningkatan akurasi yang diperoleh bersifat bertahap. Dengan demikian, penggunaan EfficientNet-B4 lebih tepat diarahkan pada skenario aplikasi yang memprioritaskan ketelitian lokalisasi dibandingkan efisiensi komputasi.

Secara keseluruhan, hasil perbandingan ini menunjukkan bahwa peningkatan kompleksitas backbone tidak selalu berbanding lurus dengan peningkatan kinerja deteksi secara menyeluruh. Dalam kondisi keterbatasan data dan durasi pelatihan, backbone dengan kompleksitas menengah seperti EfficientNet-B1 cenderung memberikan performa yang paling optimal dibandingkan backbone yang terlalu ringan maupun terlalu kompleks.

Sejalan dengan tujuan penelitian ini, yaitu menyelidiki dampak penggantian backbone terhadap kinerja YOLOv5 sebagai model deteksi objek, interpretasi hasil evaluasi tidak hanya difokuskan pada pencapaian nilai metrik yang tinggi, tetapi juga pada pola perubahan kinerja yang muncul akibat perbedaan kompleksitas backbone. Dalam konteks deteksi objek dunia nyata dengan variasi kondisi visual yang tinggi, capaian nilai Precision, Recall, dan mAP yang berada di atas 60% masih dapat dikategorikan sebagai kinerja yang baik dan representatif, sebagaimana dilaporkan pada berbagai penelitian terdahulu. Oleh karena itu, hasil evaluasi yang diperoleh pada penelitian ini dinilai telah mencerminkan kemampuan model dalam melakukan deteksi dan lokalisasi objek secara efektif, meskipun perbedaan antar backbone tidak menunjukkan peningkatan yang sangat signifikan secara numerik. Kondisi ini dapat dipengaruhi oleh keterbatasan jumlah data pelatihan, durasi training, serta sumber daya komputasi yang tersedia.

KESIMPULAN

Penelitian ini bertujuan untuk mengevaluasi secara eksperimental dampak penggantian backbone default YOLOv5 (CSPDarknet) dengan arsitektur EfficientNet terhadap kinerja sistem deteksi nilai mata uang rupiah. Fokus utama penelitian diarahkan pada analisis hubungan antara kompleksitas backbone, kinerja deteksi objek, dan efisiensi komputasi, dengan menggunakan EfficientNet-B4 sebagai backbone utama serta EfficientNet-B0 dan EfficientNet-B1 sebagai pembanding dengan kompleksitas yang lebih ringan.

Berdasarkan hasil evaluasi pada data uji, diperoleh bahwa penggunaan backbone EfficientNet-B4 pada YOLOv5 tidak menghasilkan peningkatan signifikan pada $mAP@0.5$, bahkan mengalami penurunan relatif kecil dibandingkan model YOLOv5 baseline. Namun demikian, nilai $mAP@0.5:0.95$ mengalami peningkatan, yang mengindikasikan adanya perbaikan pada ketelitian lokalisasi bounding box ketika dievaluasi pada ambang Intersection over Union (IoU) yang lebih ketat. Temuan ini

didukung oleh analisis kurva box loss yang menunjukkan penurunan loss yang lebih stabil dan konsisten pada model dengan backbone EfficientNet-B4.

Hasil perbandingan dengan EfficientNet-B0 dan EfficientNet-B1 menunjukkan bahwa backbone EfficientNet-B1, mampu memberikan keseimbangan yang lebih baik antara akurasi deteksi dan efisiensi komputasi dalam kondisi keterbatasan data, jumlah epoch pelatihan, dan sumber daya GPU. Sementara itu, backbone EfficientNet-B4 memerlukan durasi pelatihan yang lebih panjang dan dataset yang lebih besar agar kapasitas representasi fiturnya dapat dimanfaatkan secara optimal.

Implikasi praktis dari penelitian ini menegaskan bahwa pemilihan backbone pada sistem deteksi berbasis YOLOv5 harus disesuaikan dengan kebutuhan aplikasi, bukan semata-mata berdasarkan peningkatan kompleksitas arsitektur. Untuk aplikasi deteksi nilai mata uang yang menuntut efisiensi komputasi dan waktu inferensi yang cepat, backbone ringan seperti EfficientNet-B1 atau backbone default YOLOv5 direkomendasikan karena mampu memberikan kinerja deteksi yang stabil dengan beban komputasi yang lebih rendah. Sebaliknya, EfficientNet-B4 direkomendasikan untuk skenario aplikasi yang memprioritaskan ketelitian lokalisasi objek, seperti analisis visual tingkat lanjut atau proses inspeksi dengan toleransi kesalahan lokalisasi yang rendah, dengan prasyarat ketersediaan data pelatihan yang memadai dan sumber daya komputasi yang cukup.

Penelitian ini memberikan kontribusi metodologis bagi penelitian selanjutnya dalam memahami bahwa peningkatan kompleksitas backbone perlu dipertimbangkan secara cermat sesuai dengan kebutuhan aplikasi, khususnya dalam menjaga keseimbangan antara akurasi deteksi, ketelitian lokalisasi, dan efisiensi komputasi.

Penelitian selanjutnya disarankan untuk meningkatkan kinerja deteksi pada metrik mAP@0.5 melalui optimasi teknis yang terfokus, meliputi peningkatan kualitas dan konsistensi anotasi bounding box, rekalkulasi anchor box berdasarkan distribusi ukuran objek pada dataset uang rupiah, serta penyesuaian hiperparameter pelatihan seperti *learning rate*, jumlah epoch, dan strategi *fine-tuning*. Selain itu, penambahan dataset dengan citra uang kertas pada kondisi yang lebih menantang termasuk pencahayaan ekstrem, tingkat blur bervariasi, kondisi uang terlipat, dan latar belakang kompleks. Selain itu perpanjangan fase *unfreeze* pada tahap pelatihan lanjutan (Stage B) direkomendasikan untuk meningkatkan kemampuan generalisasi dan memperbaiki kinerja deteksi secara keseluruhan, khususnya pada ambang IoU standar.

Pada penelitian selanjutnya, pendekatan ini dapat dikembangkan untuk aplikasi lain yang membutuhkan ketelitian lokalisasi tinggi, seperti deteksi tumor otak pada citra MRI. Backbone EfficientNet-B4 pada penelitian ini menunjukkan kestabilan pembelajaran dan peningkatan ketelitian lokalisasi, berpotensi digunakan pada aplikasi medis yang lebih mengutamakan presisi dibandingkan kecepatan pemrosesan, dengan dukungan data dan sumber daya komputasi yang memadai.

DAFTAR REFERENSI

- Md. Jubayar Alam Rafi, M. Rony & Nazia Majadi. (2024). NSTU-BDTAKA: An open dataset for Bangladeshi paper currency detection and recognition, Elsevier
- Joseph Redmon, Santosh Divvala, Ross Girshick & Ali Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection. Available: <http://arxiv.org/abs/1506.02640>.

- Bo Xu, Bin Gao & Yunhu Li. (2024). Improved Small Object Detection Algorithm Based On Yolov5. IEEE Computer Society.
- Ardanu D Nugroho & Wiga M Baihaqi. (2023). Improved YOLOv5 with Backbone Replacement to MobileNet V3s for School Attribute Detection . SinkrOn.
- S Divya Meena, Veeramachaneni Gayathri siva sameeraja, Nagineni Sai Lasya, Meda Sathvika, Veluru Harshitha & J Sheela. (2023). Hybrid Neural Network Architecture For Multi-Label Object Recognition Using Feature Fusion. Elsevier.
- Weihai Sun, Yane Li & Hailin Feng et al. (2024). Lightweight And Accurate Aphid Detection Model Based On An Improved Deep-Learning Network. Elsevier.
- Octavian Ery Pamungkas, Puspa Rahmawati, & Dhany Maulana Supriadi et al. (2022). Classification Of Rupiah To Help Blind With The Convolutional Neural Network Method. Shinta2.
- Haiying Liu & Fengqian Sun et al. (2022). Lightweight small object detection algorithm based on improved feature fusion mode. <https://doi.org/10.3390/s22155817>.
- M. Tan and Q. V. Le. (2019). Efficientnet: Rethinking Model Scaling For Convolutional Neural Networks. Available: <http://arxiv.org/abs/1905.11946>.
- Peng Wang, Shixin Li, Weixiao Wang, Kaiwen Tang and Fankai Chen. (2024). Metal Defect Detection Models Fused EfficientNet and Involution. <https://doi.org/10.1155/js/6074853>.
- Xuecheng Li, Xiaobin Li, Biao Han, Shang Wang and Kairun Chen. (2023). Application of EfficientNet and YOLOv5 Model in Submarine Pipeline Inspection and a New Decision-Making System. MDPI.
- Alexandr Pak, Atabay Ziyaden & Kuanysh Tukeshev et al. (2020). Comparative Analysis Of Deep Learning Methods Of Detection Of Diabetic Retinopathy. <https://doi.org/10.1080/23311916.2020.1805144>.
- Andrzej D. Dobrzycki, Ana M. Bernardos and José R. Casar (2025). An Analysis of Layer-Freezing Strategies for Enhanced Transfer Learning in YOLO Architectures. MDPI.
- Prof. Dr. Siti Nurmaini and Dr. Ahmad Rizki Pratama (2023). Improving Object Detection Efficiency: Yolo V4 And Backbone Selection. AIJMSE.
- Mingxing Tan and Quoc V. Le. (2021). EfficientNetV2: Smaller Models and Faster Training. Available: <http://arxiv.org/abs/2104.00298v3>.