

**DETEKSI TANGGAL PADA KEMASAN MAKANAN MENGGUNAKAN
FASTER R-CNN DAN FPN DALAM KERANGKA DETECTRON2**
*Date Detection on Food Packaging Using Faster R-CNN and FPN in the Detectron2
Framework*

Felixiano B. Koten¹, Adriana Fanggal², Yulianto Triwahyuadi Polly³, Bertha S. Djahi⁴
Program Studi Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana
Jl. Adisucipto, Kupang, Nusa Tenggara Timur, Indonesia
Email: ¹felixianok@gmail.com, ²adrianafanggal@gmail.com, ³yuliantopolly@staf.undana.ac.id,
⁴bertha.djahi@staf.undana.ac.id

ABSTRAK.

Informasi tanggal pada kemasan makanan sangat penting untuk memastikan keamanan dan kualitas produk. Namun, variasi dalam format, jenis font, dan kondisi pencetakan menimbulkan tantangan bagi identifikasi otomatis. Studi ini bertujuan untuk mengembangkan sistem deteksi tanggal otomatis menggunakan *arsitektur Feature Pyramid Network (FPN)* dan algoritma *Faster R-CNN* yang diimplementasikan dengan kerangka kerja *Detectron2*. Model dilatih menggunakan dataset kustom yang berisi gambar kemasan makanan dari sumber primer (diambil dengan kamera) dan sumber sekunder (dari internet), dan dievaluasi menggunakan *Average Precision (AP)*, *F1-score*, dan waktu inferensi. Hasil menunjukkan kinerja yang kuat pada data sekunder, dengan rata-rata *Average Precision (mAP)* sebesar 75,9% dan *F1-score* 91,59%. Sementara itu, model ini mencapai kinerja yang cukup baik pada data primer, dengan *AP@[0.50:0.95]* sebesar 49,8% dan *F1-score* sebesar 80,42%. Hasil ini menunjukkan efektivitas penggunaan kombinasi *FPN* dan *Faster R-CNN* dalam menangani variasi kondisi kemasan. Sistem ini diharapkan dapat mendukung otomatisasi tugas inspeksi visual di industri makanan dengan secara akurat mengidentifikasi area tanggal, sehingga meningkatkan efisiensi dan keamanan dalam pengendalian kualitas produk.

Kata Kunci: Kemasan Makanan, Deteksi Tanggal, *Faster R-CNN*, *Feature Pyramid Network*, Deteksi Objek.

ABSTRACT.

Date Detection on Food Packaging Using Faster R-CNN and FPN in the Detectron2 Framework. Date information on food packaging is crucial for ensuring product safety and quality. However, variations in format, font type, and printing conditions pose challenges for automatic identification. This study aims to develop an automatic date detection system using the Feature Pyramid Network (FPN) architecture and the Faster R-CNN algorithm implemented with the Detectron2 framework. The model was trained using a custom dataset containing food packaging images from primary sources (captured with a camera) and secondary sources (from the internet), and evaluated using Average Precision (AP), F1-score, and inference time. The results show strong performance on secondary data, with an average Average Precision (mAP) of 75.9% and an F1-score of 91.59%. Meanwhile, the model achieves satisfactory performance on primary data, with AP@[0.50:0.95] of 49.8% and an F1-score of 80.42%. These results demonstrate the effectiveness of combining FPN and Faster R-CNN in handling variations in packaging conditions. This system is expected to support the automation of visual inspection tasks in the food industry by accurately identifying date areas, thereby enhancing efficiency and safety in product quality control.

Keywords: Food Packaging, Date Detection, *Faster R-CNN*, *Feature Pyramid Network*, Object Detection.

PENDAHULUAN

Informasi tanggal yang tercetak pada kemasan makanan memainkan peran penting dalam menjamin kualitas produk dan keamanan konsumen. Penandaan tanggal yang akurat membantu konsumen menentukan kesesuaian produk dan diatur secara hukum berdasarkan Undang-Undang Nomor 8 Tahun 1999 tentang Perlindungan Konsumen. Kesalahan interpretasi atau tanggal yang sulit dibaca dapat menimbulkan risiko kesehatan dan mengurangi kepercayaan konsumen. Namun, ketidakkonsistenan dalam *format*, *gaya font*, dan kualitas cetakan, khususnya penggunaan *font dot matrix*, sering membuat tanggal sulit dikenali, terutama bagi lansia atau penyandang disabilitas penglihatan.

Sebuah studi yang dilakukan oleh [1] membahas tantangan dalam mengenali informasi tanggal yang dicetak, termasuk kurangnya standar universal untuk *format* tanggal. Masalah ini timbul karena produsen individu membuat pilihan desain sendiri mengenai cara mencetak tanggal, seperti penggunaan *format* yang bervariasi, jenis huruf, dan ukuran. Akibatnya, tanggal dicetak dengan cara yang tidak konsisten sehingga sering menyulitkan untuk dibaca atau diinterpretasikan dengan benar. Misalnya, penggunaan berbagai jenis *font* dapat menyebabkan ketebalan teks yang tidak seragam, dan *font dot matrix* yang sering dipakai untuk mencetak tanggal dapat menghasilkan celah besar antar titik, membuat karakter sulit dibedakan.

Inspeksi manual terhadap label tanggal tidak hanya memakan waktu, tetapi juga rentan terhadap kesalahan manusia. Oleh karena itu, sistem otomatis yang mampu mendeteksi area tanggal secara akurat sangat diperlukan. Teknologi penglihatan komputer dan pembelajaran mesin telah menunjukkan potensi besar dalam menangani masalah deteksi dan pengenalan objek [2]. Salah satu pendekatan andal adalah kombinasi *Feature Pyramid Network* (FPN) dengan *Faster R-CNN*. FPN memungkinkan integrasi informasi dari berbagai tingkat resolusi dalam jaringan konvolusi, sehingga meningkatkan kinerja deteksi untuk objek kecil maupun besar [1]. Sementara itu, *Faster R-CNN* menggabungkan *Region Proposal Network* (RPN) dengan *Fast R-CNN* untuk menghasilkan deteksi objek yang cepat dan akurat [2]. Integrasi FPN dalam *Faster R-CNN* diharapkan dapat memperkaya representasi fitur multi-skala, sehingga meningkatkan efektivitas model dalam mengenali objek pada berbagai skala.

Studi ini bertujuan untuk mengembangkan dan mengevaluasi model deteksi tanggal dengan menggunakan arsitektur FPN dan algoritma *Faster R-CNN*, yang diimplementasikan melalui kerangka kerja Detectron2. Model dilatih dengan berbagai gambar kemasan makanan yang diambil pada jarak terkontrol (10–30 cm), dan dievaluasi

menggunakan metrik seperti AP@[0.50:0.95], AP@50, AP@75, F1-score, serta waktu inferensi. Sistem ini dirancang untuk mengenali pola pencetakan tanggal yang umum ditemukan pada berbagai jenis kemasan. Hasil yang diharapkan dari studi ini adalah mendukung otomatisasi dalam pengendalian kualitas produk, mengurangi kesalahan manusia, serta meningkatkan aksesibilitas dan keamanan bagi konsumen.

Penelitian terdahulu [3] mengusulkan model deteksi dan klasifikasi bunga berbasis NAS FPN dan *Faster R-CNN* dengan pendekatan transfer learning. Mereka menguji beberapa pretrained model seperti ResNet50, ResNet101, Inception ResNet V2, dan MobileNet V2 pada dataset *Flower 30* dan *Flower 102* yang mencakup 19.679 gambar. Hasilnya, NAS FPN dengan *Faster R-CNN* mencapai mAP 96,2% (*Flower 30*) dan 87,6% (*Flower 102*), mengungguli SSD dan *Faster R-CNN* standar. Model ini juga mendukung klasifikasi multi label, mencakup informasi botani seperti nama, klasifikasi ilmiah, dan sifat herbal bunga, yang berguna dalam bidang pertanian dan botani.

Penelitian oleh [4] mengungkapkan bahwa deteksi gulma yang cepat dan tepat sangat penting untuk mendukung proses penyiangan otomatis. Studi tersebut mengembangkan model deteksi berbasis *Faster R-CNN* yang dipadukan FPN guna meningkatkan akurasi. Hasil eksperimen menunjukkan bahwa integrasi FPN secara signifikan meningkatkan performa deteksi, terutama pada objek yang berada di latar belakang rumit.

Studi yang dilakukan oleh [1] menyatakan bahwa pengenalan tanggal kedaluwarsa pada kemasan produk merupakan tugas yang menantang karena beragamnya format tanggal, jenis huruf, serta faktor lingkungan seperti gambar buram, distorsi, dan latar belakang kompleks. Untuk mengatasi hal tersebut, mereka mengusulkan kerangka kerja berbasis *Convolutional Neural Network* (CNN) yang dirancang untuk mendeteksi dan memahami tanggal kedaluwarsa dalam 13 format berbeda.

METODOLOGI PENELITIAN

Bagian ini memaparkan kerangka metodologis dalam mengembangkan model deteksi objek untuk mengidentifikasi area tanggal pada kemasan makanan, menggunakan algoritma *Faster R-CNN* yang dikombinasikan dengan arsitektur FPN. Tahapan penelitian mencakup persiapan dataset, prapemrosesan data, perancangan dan konfigurasi model, serta evaluasi kinerja.

3.1. Dataset dan Anotasi

Dataset dalam studi ini terdiri dari 12.808 gambar RGB kemasan makanan yang memuat

informasi tanggal, dikumpulkan dari dua sumber: data primer melalui kamera smartphone dan data sekunder dari dataset publik yang sebelumnya digunakan dalam studi berbasis CNN. Untuk kompatibilitas dengan Detectron2, anotasi JSON dikonversi ke format COCO. Dataset mencakup empat label anotasi (*date*, *prod*, *due*, dan *code*), namun penelitian ini hanya difokuskan pada kelas *date*. Data dibagi dengan rasio 9:1 [5], menghasilkan 11.858 gambar untuk pelatihan dan 950 untuk pengujian, yang terdiri dari 285 gambar primer dan 665 sekunder. Gambar 1 menampilkan contoh data dengan kotak batas untuk masing-masing label.



Gambar 1. Contoh Data yang Telah Dilabeli

3.2. Preprocessing Data

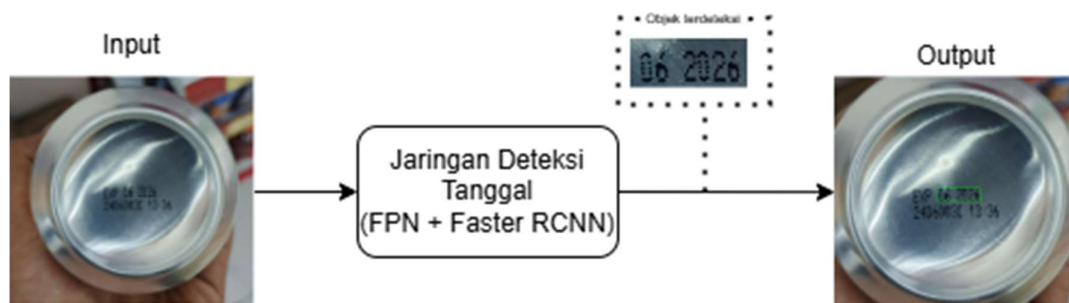
Detectron2 secara otomatis melakukan prapemrosesan sebelum gambar dimasukkan ke dalam model, termasuk mengubah sisi terpendek menjadi 800 piksel dengan menjaga rasio aspek, normalisasi kanal RGB menggunakan nilai rata-rata COCO (103.530, 116.280, 123.675) [6], Ketiga, selama proses inferensi *batch*, padding diterapkan untuk memastikan dimensi gambar yang seragam di seluruh *batch*. Langkah-langkah prapemrosesan ini penting untuk menjaga konsistensi dalam *pipeline input* model dan meningkatkan konvergensi selama pelatihan.

Selain itu, pada tahap pelatihan model, augmentasi data dilakukan secara otomatis melalui konfigurasi *DatasetMapper* dalam *Detectron2*. Teknik augmentasi ini mencakup dua transformasi utama, yaitu *Resize* dan *RandomFlip*. Penyesuaian ukuran objek melalui proses resizing telah terbukti efektif dalam meningkatkan kemampuan sistem untuk mendeteksi objek dengan ukuran bervariasi[7]. Teknik *resize* yang digunakan

dalam penelitian ini adalah *ResizeShortestEdge* akan secara acak memilih salah satu nilai dari rentang (640, 672, 704, 736, 768, 800) sebagai ukuran sisi terpendek gambar, dengan batas maksimal tinggi atau lebar tidak melebihi 1333 piksel [8]. Sementara itu, *RandomFlip* digunakan untuk membalik gambar secara horizontal dengan probabilitas tertentu. Proses augmentasi data seperti rotasi dan flipping digunakan untuk memperkaya variasi dalam dataset, yang bertujuan agar model mampu mengenali objek dari berbagai sudut dan kondisi pencahayaan. Strategi ini efektif dalam meningkatkan akurasi deteksi objek karena mencerminkan kemungkinan situasi nyata dalam data uji[9]. Kedua teknik augmentasi ini bertujuan untuk meningkatkan keragaman data pelatihan serta memperkuat kemampuan generalisasi model terhadap variasi posisi dan ukuran objek tanggal pada kemasan makanan[10].

3.3. Arsitektur Model

Penelitian ini menggunakan arsitektur model yang mengadaptasi *framework* Detectron2 dengan pendekatan *Faster* R-CNN yang terintegrasi dengan FPN. Model memanfaatkan *backbone* ResNet50 yang telah melalui pra-pelatihan pada dataset ImageNet, kemudian dikombinasikan dengan FPN untuk mengekstraksi fitur dari berbagai skala secara lebih efisien. Konfigurasi model merujuk langsung pada arsitektur pra-latih “COCO-Detection/*Faster* R-CNN_R_50_FPN_3x.yaml” dari model zoo Detectron2, sehingga seluruh komponen utama seperti *backbone*, RPN, dan ROI (*Region of Interest*) *Heads* telah disesuaikan secara optimal mengikuti standar COCO. Secara keseluruhan, alur proses deteksi tanggal ditampilkan pada Gambar 2.



Gambar 2. Alur Deteksi Tanggal

Backbone

Model ini memanfaatkan ResNet50 yang dipadukan dengan FPN sebagai *backbone* untuk mengekstraksi fitur dari citra input. Pada tahap awal, ResNet50 bertugas menghasilkan fitur dasar dari gambar, sementara FPN menggabungkan fitur dari beberapa tingkat pada ResNet menggunakan jalur *top-down* dan koneksi lateral, menghasilkan *feature map* dengan

berbagai resolusi [11].

Struktur ResNet (Bottom-Up)

ResNet terdiri dari beberapa tahap utama, yaitu:

- Stem: Lapisan awal untuk mengolah citra input.
 - ❖ Conv2d(3, 64, *kernel_size*=7, *stride*=2, *padding*=3) mengubah gambar RGB menjadi 64 kanal fitur.
 - ❖ FrozenBatchNorm2d(64) digunakan untuk normalisasi batch guna meningkatkan stabilitas pelatihan.
- Blok *Residual* (res2 hingga res5): Setiap blok terdiri dari beberapa unit *BottleneckBlock*, dengan kombinasi konvolusi 1×1 , 3×3 , dan 1×1 untuk menangkap informasi dari berbagai tingkat. *Shortcut connection* digunakan agar informasi dari lapisan sebelumnya tetap terjaga. Tabel 1 menunjukkan konfigurasi tiap blok ResNet50.

Tabel 1. Proses ResNet50

Blok	Input Channel	Output Channel	Stride
Res2	64	256	1
Res3	256	512	2
Res4	512	1024	2
Res5	1024	2048	2

Feature Pyramid Network (FPN)

FPN bertugas menghasilkan fitur dari berbagai skala dengan cara:

- *Lateral Connection* menggunakan konvolusi 1×1 (*fpn_lateral*) untuk mereduksi dimensi saluran fitur dari *backbone*.
- *Top-down Pathway* menggunakan konvolusi 3×3 (*fpn_output*) untuk menghasilkan fitur akhir yang siap digunakan oleh detektor objek.

Tabel 2. Proses FPN

FPN Level	Lateral(1×1)	Output(3×3)
P2	$256 \rightarrow 256$	$256 \rightarrow 256$
P3	$512 \rightarrow 256$	$256 \rightarrow 256$
P4	$1024 \rightarrow 256$	$256 \rightarrow 256$
P5	$2048 \rightarrow 256$	$256 \rightarrow 256$
P6	MaxPool dari P5	-

Berdasarkan Tabel 3 FPN menggabungkan *feature map* dari ResNet (C2 hingga C5) dengan mekanisme *upsampling*, menghasilkan level P2 hingga P5. P6 dihasilkan melalui konvolusi tambahan dengan *stride* 2 dari P5.

Contoh bagaimana FPN menghasilkan *feature map* untuk input berukuran 480×640, prosesnya dijelaskan dalam Tabel 3 dan 4.

Tabel 3. Contoh ResNet Pada Inputan

<i>Stage</i>	<i>Layer</i>	<i>Ukuran Feature Map</i>	<i>Channel</i>
<i>Input</i>	-	480×640	3
<i>Stem Conv</i>	7×7 Conv, <i>Stride</i> =2	240×320	64
<i>MaxPool</i>	3×3 MaxPool, <i>Stride</i> =2	120×160	64
<i>Res2</i>	<i>Bottleneck</i> $\times 3$	120×160	256
<i>Res3</i>	<i>Bottleneck</i> $\times 4$	60×80	512
<i>Res4</i>	<i>Bottleneck</i> $\times 5$	30×40	1024
<i>Res5</i>	<i>Bottleneck</i> $\times 6$	15×20	2048

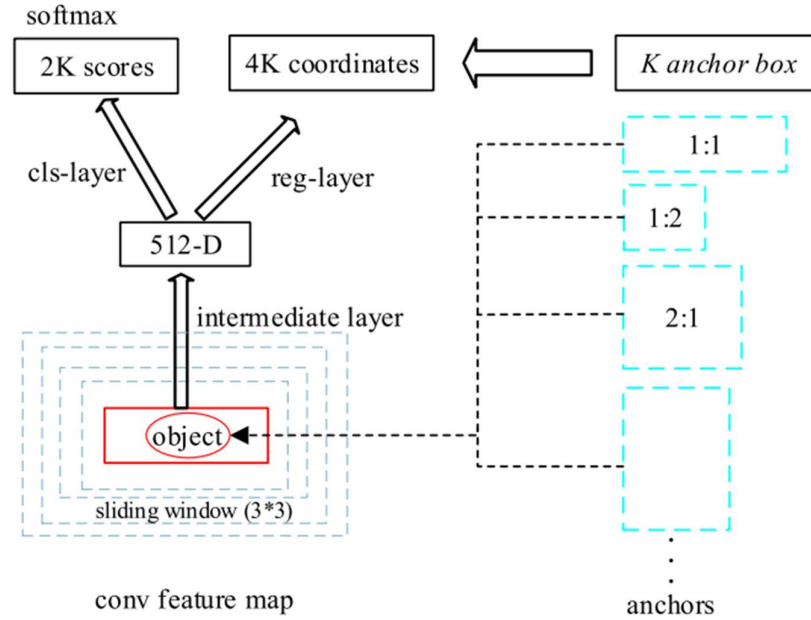
Tabel 4. Contoh FPN Pada Inputan

<i>Pyramid Level</i>	<i>Input dari ResNet</i>	<i>Ukuran Feature Map</i>	<i>Channel</i>
P2	Dari Res2 (C2)	120×160	256
P3	Dari Res3 (C3)	60×80	256
P4	Dari Res4 (C4)	30×40	256
P5	Dari Res5 (C5)	15×20	256
P6	Dari P5 (dengan 3×3 stride 2 Conv)	8×10	256

Melalui proses ini, dihasilkan sekumpulan *feature maps* dari berbagai tingkat resolusi (P2–P6), yang mencerminkan kombinasi informasi visual detail dan konteks semantik dari gambar input.

Region Proposal Network (RPN)

Komponen berikutnya adalah RPN, yang memanfaatkan peta fitur dari FPN untuk menghasilkan sejumlah wilayah kandidat (region proposals) yang kemungkinan besar mengandung objek. RPN bekerja dengan menggunakan sebuah kepala jaringan konvolusional yang menghitung skor objectness dan memprediksi pergeseran posisi anchor (anchor deltas) pada setiap titik dalam peta fitur [12]. Sistem anchor yang digunakan mengikuti konfigurasi default, dengan ukuran dan rasio aspek yang telah disesuaikan untuk mendeteksi objek secara efisien [13]. Struktur rinci dari RPN ditampilkan pada Gambar 3.



Gambar 3. Struktur RPN

Setelah melewati tahap RPN, proses dilanjutkan oleh *ROI Heads*, yang bertugas memproses area kandidat menggunakan metode ROI Align. Fitur yang diambil dari setiap region proposal kemudian dimasukkan ke dalam jaringan *fully-connected* (*box head*) untuk melakukan klasifikasi objek serta regresi terhadap koordinat bounding box. Komponen *box predictor* pada *ROI Heads* menghasilkan skor kelas dan koordinat akhir dari bounding box, sehingga deteksi objek dapat dilakukan dengan tingkat akurasi yang tinggi.

3.4. Konfigurasi Pelatihan

Konfigurasi pelatihan dalam penelitian ini dirancang secara khusus dengan sejumlah penyesuaian penting guna meningkatkan stabilitas dan performa model. Arsitektur dasar yang digunakan berasal dari konfigurasi bawaan Detectron2, yaitu **Faster R-CNN R50-FPN 3x** dari model zoo. Dataset kustom telah diregistrasi ke dalam format Detectron2, baik untuk data pelatihan maupun pengujian.

Model diinisialisasi menggunakan bobot pra-latih dari COCO, dengan sejumlah parameter pelatihan utama yang disesuaikan, antara lain ukuran *batch* (IMS_PER_BATCH) sebesar 8 dan jumlah iterasi maksimum (MAX_ITER) sebanyak 30.000. Nilai *learning rate* awal ditetapkan sebesar 0,001, dengan tahap pemanasan (*warm-up*) selama 1.000 iterasi.

Untuk mengontrol penurunan learning rate secara lebih adaptif, digunakan *scheduler CosineAnnealingWarmRestarts* dari PyTorch, dengan parameter $T_0 = 2000$ dan $\eta_{min} = 1e-6$, yang diimplementasikan melalui *override* terhadap metode

build_lr_scheduler.

Selama pelatihan, dua *custom hook* digunakan:

- EvalHookAtIterations, yang menjalankan evaluasi pada iterasi tertentu seperti 16000, 18000, hingga 25000 dengan menggunakan evaluator COCO.
- SaveModelHook, yang menyimpan model pada iterasi yang telah ditentukan serta menyimpan model akhir (*model_final*) setelah pelatihan selesai.

Konfigurasi tambahan juga dilakukan untuk mengatur jumlah kelas menjadi 4, menyesuaikan *hyperparameter* pada RPN (seperti ambang NMS dan jumlah anchor), serta menetapkan skor minimum 0.5 untuk tahap inferensi.

HASIL DAN PEMBAHASAN

4.1. Evaluasi Performa Model

Penelitian ini memanfaatkan arsitektur *Faster R-CNN* dengan *Feature Pyramid Network* (FPN) yang diimplementasikan menggunakan *framework* Detectron2. Model dilatih pada dataset sekunder dan primer yang telah dikonversi ke format COCO dan diregistrasikan ke dalam *pipeline* pelatihan Detectron2. Strategi pelatihan melibatkan *learning rate scheduler* CosineAnnealingWarmRestarts serta penggunaan *custom hook* untuk evaluasi dan penyimpanan model pada iterasi tertentu. Evaluasi model dilakukan pada beberapa titik iterasi pelatihan. Berdasarkan hasil evaluasi, performa terbaik diperoleh pada iterasi ke-25000, sehingga model hasil pelatihan pada iterasi ini dipilih sebagai model akhir. Evaluasi menggunakan metrik standar deteksi objek seperti AP@[0.50:0.95], AP@50, AP@75, F1-score, serta waktu inferensi rata-rata per gambar.

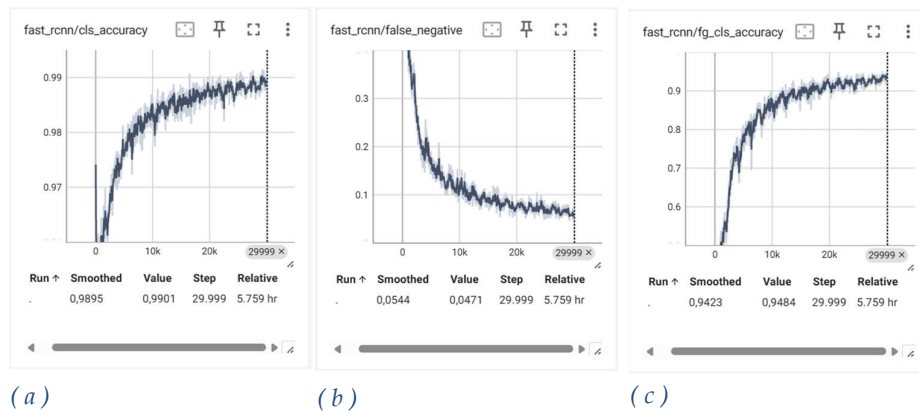
Tabel 5. Evaluasi Performa Model

Metrik	Data Sekunder	Data Primer
AP@[0.50:0.95]	75.9%	49.8%
AP@50	94.60%	80.2%
AP@75	88.70%	52.9%
F1-score	91.59%	80.42%
Waktu Inferensi (avg)/gambar	0.309059 detik	0.304 detik
Jumlah citra terdeteksi (IoU:0.5)	644 citra	280 citra

Berdasarkan Tabel 5, model mampu menunjukkan kinerja deteksi yang sangat baik pada dataset sekunder, ditandai dengan nilai *Average Precision* (AP) pada ambang 50% (AP@50) yang mencapai 94,60%. Hasil ini mencerminkan bahwa model memiliki

kemampuan yang tinggi dalam mengidentifikasi objek target pada gambar dengan kualitas yang relatif seragam. Di sisi lain, terjadi penurunan performa saat model diuji pada dataset primer. Penurunan ini dapat dikaitkan dengan tingkat kompleksitas visual yang lebih tinggi serta ketidakkonsistenan kualitas gambar, mengingat data primer dikumpulkan menggunakan berbagai jenis kamera ponsel yang memiliki spesifikasi berbeda-beda. Meskipun begitu, nilai *F1-score* pada kedua jenis data tetap tergolong tinggi, yang menandakan bahwa model mampu menjaga keseimbangan yang baik antara tingkat presisi (*precision*) dan sensitivitas (*recall*). Selain itu, waktu inferensi yang cenderung konsisten selama pengujian memperlihatkan bahwa sistem bekerja secara efisien dalam melakukan proses deteksi, tanpa mengalami keterlambatan signifikan.

Selain evaluasi berbasis metrik umum seperti AP dan *F1-score*, proses pelatihan model juga dimonitor menggunakan *TensorBoard* untuk mengevaluasi stabilitas dan efektivitas pelatihan. Tiga metrik penting yang diamati adalah *cls_accuracy*, *false_negative*, dan *fg_cls_accuracy*, sebagaimana ditampilkan pada Gambar 4.



Gambar 4.

Evaluasi performa pelatihan model Faster R-CNN + FPN menggunakan dataset *sekunder*.

(a) Akurasi klasifikasi (*cls_accuracy*), (b) Tingkat kesalahan deteksi (*false_negative*), dan (c) Akurasi klasifikasi objek target (*fg_cls_accuracy*)

Gambar 4 menunjukkan tren pelatihan model saat diuji pada dataset sekunder, di mana akurasi klasifikasi (*cls_accuracy*) meningkat secara konsisten dan mencapai nilai akhir sebesar 0.9901. Ini menunjukkan bahwa model mampu mengklasifikasikan proposal wilayah secara akurat. Nilai *false negative* menurun tajam dari lebih dari 0.3 menjadi 0.0471, menandakan bahwa model semakin jarang gagal mendeteksi objek yang relevan.

Adapun akurasi klasifikasi untuk kelas *foreground* (*fg_cls_accuracy*) meningkat stabil hingga mencapai 0.9484, menunjukkan bahwa model dapat mengenali objek tanggal dengan presisi tinggi.

KESIMPULAN DAN SARAN

Studi ini menerapkan model deteksi objek untuk mengidentifikasi tanggal pada kemasan makanan menggunakan arsitektur *Faster R-CNN* yang terintegrasi dengan *Feature Pyramid Network* (FPN). Hasil eksperimen menunjukkan bahwa model ini berkinerja baik pada dataset sekunder, dengan rata-rata *Average Precision* (mAP) sebesar 75,9% dan skor F1 sebesar 91,59%. Namun, kinerja menurun pada dataset utama yang lebih mencerminkan kondisi dunia nyata, dengan $AP@[0.50:0.95]$ sebesar 49,8% dan skor F1 sebesar 80,42%. Penurunan ini disebabkan oleh kompleksitas visual seperti adanya kode produksi yang mirip dengan tanggal dan kedekatan tanggal dengan elemen kemasan lainnya. Meskipun demikian, model menunjukkan tingkat keberhasilan deteksi rata-rata yang tinggi sebesar 92% pada berbagai format tanggal dan mempertahankan waktu inferensi efisien di bawah 0,31 detik per gambar. Temuan ini mengonfirmasi potensi penggabungan FPN dengan *Faster R-CNN* untuk deteksi tanggal, sekaligus menyoroti kebutuhan akan perbaikan lebih lanjut dalam menangani skenario dunia nyata yang kompleks.

Untuk pengembangan selanjutnya, disarankan agar data pelatihan diperluas dengan menambahkan kumpulan data lokal dari wilayah tempat penelitian dilaksanakan guna meningkatkan generalisasi model terhadap variasi visual dunia nyata. Selain itu, sistem ini dapat disempurnakan dengan menambahkan kemampuan *Optical Character Recognition* (OCR) agar informasi tanggal yang terdeteksi dapat langsung dikonversi menjadi teks digital yang dapat digunakan secara praktis oleh pengguna akhir.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada rekan tim dan kolaborator atas dukungan, diskusi, serta bantuan teknis yang diberikan, serta mengakui peran penting Detectron2 dari *Facebook AI Research* sebagai kerangka kerja sumber terbuka dalam implementasi model penelitian ini..

DAFTAR PUSTAKA

- [1] A. C. Seker and S. C. Ahn, "A generalized framework for recognition of expiration

- dates on product packages using fully convolutional networks,” *Expert Syst. Appl.*, vol. 203, p. 117310, Oct. 2022, doi: 10.1016/j.eswa.2022.117310.
- [2] Q. Feng, X. Xu, and Z. Wang, “Deep learning-based small object detection: A survey,” *Math. Biosci. Eng.*, vol. 20, no. 4, pp. 6551–6590, 2023, doi: 10.3934/mbe.2023282.
 - [3] I. Patel and S. Patel, “An optimized deep learning model for flower classification using NAS-FPN and faster R-CNN,” *Int. J. Sci. Technol. Res.*, vol. 9, no. 03, pp. 5308–5318, 2020.
 - [4] Y. Mu *et al.*, “A Faster R-CNN-Based Model for the Identification of Weed Seedling,” *Agronomy*, vol. 12, no. 11, p. 2867, Nov. 2022, doi: 10.3390/agronomy12112867.
 - [5] Baiq Nurul Azmi, Arief Hermawan, and Donny Avianto, “Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver,” *JTIM J. Teknol. Inf. dan Multimed.*, vol. 4, no. 4, pp. 281–290, Feb. 2023, doi: 10.35746/jtim.v4i4.298.
 - [6] admin, “detectron2.data.transforms.” [Online]. Available: https://detectron2.readthedocs.io/en/latest/modules/data_transforms.html?utm_source=chatgpt.com
 - [7] F. A. Rafi, A. Fanggidae, and Y. T. Polly, “Asphalt Road Damage Detection System Using Canny Edge Detection,” *J. Komput. dan Inform.*, vol. 11, no. 1, pp. 85–90, Mar. 2023, doi: 10.35508/jicon.v11i1.10100.
 - [8] C. Zhang, T.-Y. Pan, T. Chen, J. Zhong, W. Fu, and W.-L. Chao, “Learning with free object segments for long-tailed instance segmentation,” in *European Conference on Computer Vision*, Springer, 2022, pp. 655–672.
 - [9] M. Michael, “Identifikasi Kendaraan Beroda Menggunakan Algoritma Yolov5,” 2024, *Jurnal Buana Informatika*.
 - [10] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, “Image data augmentation for deep learning: A survey,” *arXiv Prepr. arXiv2204.08610*, 2022.
 - [11] P. Potrimba, “What is ResNet-50?” [Online]. Available: <https://blog.roboflow.com/what-is-resnet-50/>
 - [12] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 658–666, 2019, doi: 10.1109/CVPR.2019.00075.
 - [13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.