

IMPLEMENTASI ALGORITMA FASTER R-CNN DALAM DETEKSI AKTIVITAS MEROKOK DI LINGKUNGAN KAMPUS

Nazla Abay Daud Istanto^{1*}, Bagus Satrio Waluyo Poetro²

^{1,3}Universitas Islam Sultan Agung, Indonesia

abayistanto@gmail.com^{1*} bagusswp@unissula.ac.id²

Received: 10-07- 2027

Revised: 20-07-2025

Approved: 30-07-2025

ABSTRAK

Penerapan pengaturan kawasan tanpa rokok di lingkungan kampus menghadapi tantangan dalam hal pengawasan dan penegakan. Penelitian ini bertujuan untuk mengetahui performa algoritma deteksi objek berbasis deep learning, yaitu Faster R-CNN, dalam mendeteksi aktivitas merokok di area kampus Universitas Islam Sultan Agung (UNISSULA). Dataset yang digunakan terdiri dari 1935 gambar beranotasi aktivitas merokok yang diperoleh dari Roboflow, dengan pembagian data 85% pelatihan dan 15% validasi. Model dilatih menggunakan Google Colab dan diuji berdasarkan metrik evaluasi seperti accuracy, precision, recall, dan f1-score. Hasil mengetahui bahwa Faster R-CNN mempunyai performa yang lebih unggul dengan nilai evaluasi terbaik mencapai 100% pada threshold 0.5. Temuan ini menyimpulkan bahwa Faster R-CNN cocok untuk digunakan dalam sistem pendeteksian aktivitas merokok di lingkungan kampus, terutama dalam konteks akurasi dan konsistensi deteksi.

Kata kunci: Deteksi, Faster R-CNN, Computer Vision, Deep Learning, Kampus Bebas Rokok.

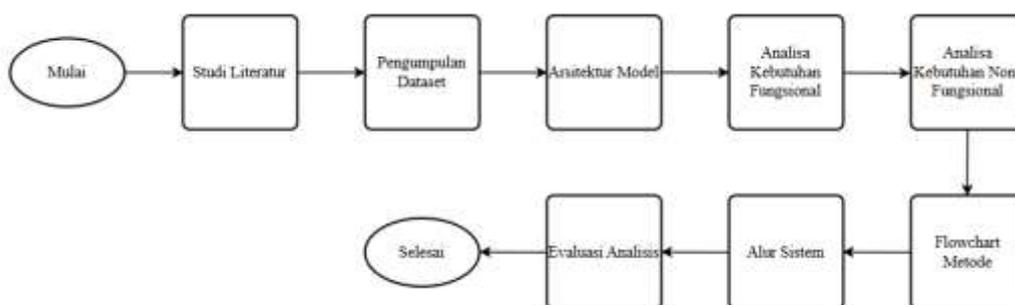
PENDAHULUAN

Kebijakan larangan merokok telah diterapkan di berbagai kampus untuk menciptakan lingkungan yang sehat dan nyaman bagi seluruh civitas akademik. Banyak sekali tulisan-tulisan reklame yang berisi larangan merokok, baik bagi mahasiswa, karyawan, maupun dosen ketika berada di lingkungan kampus (Sudarto, 2020). Merokok di tempat umum, khususnya di lingkungan kampus, dapat menyebabkan berbagai dampak buruk, mengingat asap rokok mengandung campuran daun tembakau kering yang dicacah dan dicampur dengan bahan-bahan lain, seperti perekat dan senyawa kimia yang memberikan dampak buruk terhadap kesehatan perokok, termasuk perokok pasif yang tidak sengaja menghisap asap rokok tersebut (Tivany Ramadhani et al., 2023). Dengan laju perubahan teknologi yang semakin pesat dalam kehidupan masyarakat serta kemampuan komputasi komputer yang semakin tinggi. Salah satu area teknologi yang sedang naik daun saat ini adalah *machine learning* yang mana merupakan bagian dari ilmu kecerdasan buatan atau yang dikenal sebagai *artificial intelligence (AI)*. Salah satu cabang dari *AI* adalah *deep learning* (Ramadhanu et al., 2023). dan dapat melakukan pembelajaran yang lebih mendalam sehingga menjadikan *deep learning* sebagai inti dari kemajuan *AI* serta memungkinkan untuk mendeteksi objek. Di bidang *deep learning*, menjadi populer dan metode ini tidak mengubah kualitas gambar tetapi mengubah dimensi. deteksi tambahan memakai *CNN* dengan metode pembelajaran transfer. Keunggulan metode ini yaitu kemampuan untuk melakukan pra-pelatihan menggunakan informasi yang dipelajari dari model *CNN* yang telah dilatih (Poetro et al., 2024). *Faster RCNN* merupakan salah satu metode *deep learning* yang digunakan untuk mengenali suatu objek pada citra. Pengenalan dilakukan dengan menelusuri ciri-ciri yang dimiliki oleh objek pada citra. Penelusuran dilakukan melalui sejumlah *layer* melalui proses konvolusi atau yang lebih dikenal dengan nama *Convolutional Neural Network (CNN)* (Alamsyah & Pratama, 2019).

Pada penelitian sebelumnya Sistem dibangun menggunakan metode *Faster R-CNN* untuk mengetahui keberadaan objek kereta api antara lokomotif dan gerbong berdasarkan tingkat cahaya dan jarak terhadap objek. Sistem memperoleh keakurasian deteksi terhadap objek rata-rata akurasi deteksi objek lokomotif sebesar 86,40%, dan rata-rata akurasi deteksi objek gerbong sebesar 97,23% (Pardede & Hardiansah, 2022). Pada penelitian sebelumnya meneliti akurasi menggunakan model *Faster R-CNN* yang dilatih dalam mendeteksi senjata genggam kelas pisau. Hasil akhir dari proses pengembangan model *Faster R-CNN* ini adalah model yang berhasil mendeteksi senjata genggam kelas pisau dengan akurasi sebanyak 87%, hasil akurasi didapatkan dari pengujian terhadap 475 gambar digital yang dilakukan di *Google Colab* (Pradana et al., 2022). Pada penelitian sebelumnya pembuatan sistem pendeteksian objek ini meliputi tahap pelatihan untuk membentuk model *Faster R-CNN*. Berdasarkan hasil implementasi dan uji coba, disimpulkan bahwa sistem ini dapat mengklasifikasikan jenis hewan karnivora dan herbivora dengan total *loss model* hasil pelatihan berada pada 0.06 dan rata-rata tingkat akurasi sebesar 89%. Hasil perhitungan *Recall* dan *Precision* menunjukkan bahwa performa dari sistem klasifikasi yang dihasilkan sangat baik, dengan nilai *Recall* 100% terdapat pada kelompok citra hewan Cheetah, Eagle, Komodo, Shark, Tiger, Bull, Guineapig, dan Zebra (Putri & Fahrurozi, 2022). Pada penelitian sebelumnya Sistem ini menggunakan permodelan *Faster Regional Convolutional Neural Network* yang dibuat pada *Tensorflow* dengan memprosesnya di mini komputer atau Raspberry Pi 4B. Hasil akurasi pada sistem ini sebesar 0.9025 atau sebesar 90.25% dengan rata-rata waktu komputasi didalam *Raspberry Pi 4B* sebesar 7.638 detik per gambar (Audina et al., 2021). Meskipun metode *Faster r-cnn* telah terbukti berhasil dalam berbagai deteksi, penelitian performa mereka dalam konteks deteksi aktivitas merokok di lingkungan kampus masih terbatas. Oleh karena itu, penelitian ini bertujuan untuk mengetahui performa dari *Faster R-CNN* dalam mendeteksi aktivitas merokok, dengan harapan dapat memberikan wawasan yang lebih baik mengenai metode untuk diterapkan dalam pengawasan kesehatan di lingkungan Kampus.

METODE PENELITIAN

Adapun penelitian yang digunakan oleh penelitian tersebut, yaitu sebagai berikut :



Gambar 1. Alur flowchart penelitian

1. Studi Literatur

Peneliti mengumpulkan beberapa *e-book*, artikel, jurnal, skripsi dan tesis hasil penelitian yang telah dilakukan oleh peneliti sebelumnya, dan juga beberapa laman di internet sebagai media untuk peneliti mempelajari terkait *Machine Learning*, *Deep Learning*, *Computer vision*, dan *Faster R-CNN* untuk membantu peneliti menyelesaikan

penelitian.

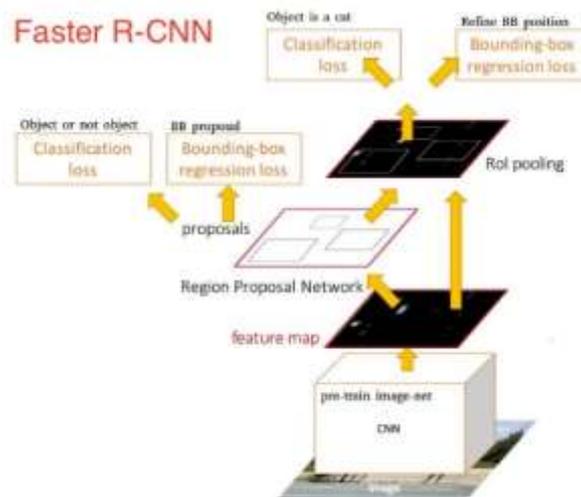
2. Pengumpulan Dataset



Gambar 2. Pengumpulan Dataset

Pada Gambar 2. Dataset yang digunakan dalam penelitian ini terdiri dari kumpulan gambar rokok dengan variasi dalam pencahayaan, sudut pengambilan gambar, dan kondisi. Data ini dapat diperoleh dari dataset publik seperti Roboflow. Dataset yang digunakan berjenis *Coco (Common Objects in Context)*.

3. Arsitektur Model



Gambar 3. Arsitektur *faster r-cnn*

Pada Gambar 3. tahapan model *Faster R-CNN*, gambar pertama-tama akan diproses oleh *convolutional neural network (CNN)*. *CNN* akan mencari pola-pola atau fitur seperti tepi, sudut, tekstur, atau bentuk di dalam *dataset*. *CNN* akan menghasilkan sebuah "peta fitur" (*feature map*), yang bisa dianggap sebagai gambaran kasar dari gambar, namun dengan informasi yang lebih fokus pada objek yang ada di dalamnya. Setelah gambar diubah menjadi peta fitur, ada bagian penting lain yang disebut *Region Proposal Network (RPN)*. Fungsi *RPN* adalah untuk mencari kemungkinan lokasi objek dalam gambar. Artinya, *RPN* akan mencoba untuk menemukan **area-area** di dalam

gambar yang berpotensi berisi objek rokok. *RPN* bekerja dengan cara membuat banyak *bounding box* (kotak pembatas) di seluruh gambar dan menilai mana yang paling mungkin berisi objek yang ingin dideteksi. Setelah *RPN* menentukan beberapa area yang mungkin berisi objek, langkah berikutnya adalah *RoI Pooling*. Di bagian ini, gambar yang telah diproses oleh *RPN* akan dipotong-potong sesuai dengan area yang dianggap penting. *RoI Pooling* mengambil potongan-potongan gambar ini dan mengubahnya menjadi ukuran yang sama (misalnya 7x7 piksel). Dengan cara ini, model bisa memproses setiap area (potongan gambar) yang berisi objek dengan cara yang sama, sehingga model bisa lebih efisien dalam membuat prediksi. Pada tahap terakhir, *Faster R-CNN* menggunakan dua Langkah yaitu *Klasifikasi* dan *Regresi*. *Klasifikasi*, model akan menentukan apakah setiap area (potongan gambar) yang diambil oleh *RPN* berisi objek rokok atau tidak. Ini adalah bagian dari proses pengenalan objek. *Regresi*, selain mengklasifikasikan objek, model juga akan memperbaiki lokasi objek dalam gambar. Ini dilakukan dengan menghitung ulang posisi dan ukuran *bounding box* untuk mendapatkan prediksi lokasi objek yang lebih akurat.

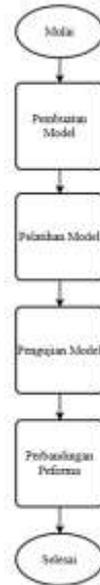
4. Analisa Kebutuhan Fungsional

Pada tahap ini kebutuhan fungsional merupakan proses identifikasi dan informasi yang akan dihasilkan oleh sistem yang akan dibangun.

5. Analisa Kebutuhan Non-Fungsional

Pada tahap ini kebutuhan non fungsional adalah kebutuhan yang dimiliki setiap sistem yang membatasi pengembangan ataupun standarisasi sistem yang akan dibuat. Kebutuhan tersebut terdiri dari :

1. Perangkat Keras (*Hardware*) yang digunakan dalam pembuatan aplikasi sebagai berikut :
 - a. Laptop Toshiba C655, *Prosesor* : Intel(R) Core (TM) i3, *RAM* : 4,00 GB, dan *SSD* : 256 GB
 2. Perangkat Lunak (*Software*) *Software* yang digunakan dalam perancangan dan pembuatan aplikasi ini sebagai berikut:
 - a. Sistem Operasi : *Windows 10*
 - b. Bahasa Pemograman : *Python*
 - c. *Editor/Compile* : *Google Colab*
 - d. Perancangan : *Visual Paradigm*.
6. *Flowchart* Metode



Gambar 4. Alur *flowchart* model

a. Pembuatan model

Pada tahap pembuatan model, model deteksi berbasis *Faster R-CNN* dirancang dan dikembangkan. Proses ini melibatkan pengumpulan *dataset* berupa gambar atau video yang relevan dengan aktivitas merokok, dilanjutkan dengan anotasi *dataset* untuk menandai area-area yang mengindikasikan aktivitas merokok. Pembuatan model ini bertujuan agar algoritma dapat mengenali objek rokok, asap, dan aktivitas merokok dengan akurasi tinggi.

b. Pelatihan model

Setelah model selesai dirancang, tahap berikutnya adalah melatih algoritma menggunakan *dataset* yang telah disiapkan. Proses pelatihan melibatkan pemberian data *input* kepada algoritma untuk mengajarkan bagaimana mendeteksi pola-pola tertentu yang terkait dengan aktivitas merokok. Pada tahap ini, berbagai parameter seperti *learning rate*, jumlah *epoch*, dan ukuran *batch* dioptimalkan untuk memastikan algoritma dapat belajar dengan baik. *Faster R-CNN* dilatih untuk melihat performanya.

c. Pengujian model

Tahap pengujian dilakukan setelah proses pelatihan selesai. Pada tahap ini, model yang telah dilatih diuji menggunakan data uji yang belum pernah dilihat sebelumnya oleh algoritma. Tujuannya adalah untuk mengevaluasi kemampuan model dalam mendeteksi aktivitas merokok di berbagai kondisi, seperti pencahayaan, sudut kamera, dan tingkat keramaian. Hasil pengujian mencakup metrik seperti *Precision*, *Recall*, *Average Precision (AP)*, dan *F1-Score*.

d. Perbandingan performa

Hasil dari pengujian kedua algoritma dibandingkan untuk menganalisis kelebihan dan kekurangannya masing-masing. *Faster R-CNN* biasanya lebih unggul dalam hal akurasi deteksi, tetapi memiliki waktu inferensi yang lebih lama. Analisis ini bertujuan untuk menentukan algoritma yang paling sesuai untuk diterapkan di lingkungan Unissula, terutama untuk mendukung upaya penegakan kebijakan zona larangan merokok.

7. Alur Sistem



Gambar 5. Alur Sistem

Gambar 5. menjelaskan alur kerja sistem yang dimulai dengan tahap inisialisasi proses. Data utama berupa rekaman video aktivitas merokok di lingkungan Unissula digunakan sebagai input sistem. Video tersebut diproses oleh algoritma *Faster R-CNN*, untuk mendeteksi aktivitas merokok. Hasil deteksi dari kedua algoritma ini kemudian dibandingkan melalui analisis performa yang mencakup *Precision*, *Recall*, *Average Precision (AP)*, dan *F1-Score*. Setelah proses analisis selesai, hasil penelitian dirangkum untuk menentukan algoritma yang lebih unggul, dan penelitian pun diakhiri.

8. Evaluasi dan Analisis

Mengevaluasi untuk melihat apakah sistem yang dirancang bekerja seperti yang diharapkan dengan melihat hasil pengukuran metrik seperti *Precision*, *Recall*, *Average Precision (AP)*, dan *F1-Score*.

a. *Precision*

Precision adalah salah satu metrik evaluasi dalam *machine learning*, terutama pada masalah klasifikasi, untuk mengukur keakuratan prediksi positif model. *Precision* menunjukkan proporsi prediksi positif yang benar-benar relevan dibandingkan dengan seluruh prediksi positif yang dibuat oleh model. *Precision* dihitung dengan rumus :

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

- TP: *True Positives* adalah Jumlah prediksi positif yang benar (model memprediksi positif, dan sebenarnya memang positif).
- FP: *False Positives* adalah Jumlah prediksi positif yang salah (model memprediksi positif, tetapi sebenarnya negatif).

b. *Recall*

Recall adalah metrik evaluasi dalam *machine learning* yang digunakan untuk mengukur seberapa baik model mendeteksi semua kasus positif yang sebenarnya ada. *Recall* sering digunakan untuk menilai kemampuan model dalam menangkap semua *True Positive (TP)* dari data yang tersedia. *Recall* dihitung dengan rumus :

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

- TP: *True Positives* adalah jumlah prediksi positif yang benar (model memprediksi positif, dan sebenarnya memang positif).
- FN: *False Negatives* adalah Jumlah prediksi negatif yang salah (model memprediksi negatif, tetapi sebenarnya positif).

c. *Average Precision (AP)*

Average Precision (AP) adalah metrik evaluasi yang digunakan dalam *machine learning*, terutama untuk tugas ***object detection*** dan ***information retrieval***. *Average Precision* mengukur kinerja model dengan mempertimbangkan baik ***precision*** maupun ***recall***. *Average Precision* dihitung dengan rumus :

$$AP = \sum_{n=1}^N (R_n - R_{n-1}) P_n$$

- R_n : Nilai *recall* pada titik ke-n dari *Precision-Recall Curve*.
- $R_n - R_{n-1}$: Perbedaan nilai *recall* antara dua titik berturut-turut.
- P_n : Nilai *precision* pada titik ke-n dari *Precision-Recall Curve*.

d. *F1-Score*

F1-Score adalah metrik evaluasi dalam *machine learning* yang digunakan untuk mengukur keseimbangan antara *precision* dan *recall*. *F1-Score* sering digunakan ketika penting untuk mempertimbangkan kedua aspek tersebut secara bersamaan, terutama dalam kasus data tidak seimbang. *F1-Score* dihitung dengan rumus :

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

HASIL DAN PEMBAHASAN

Pada model deteksi aktivitas merokok ini dibuat dengan beberapa parameter yang sebelumnya sudah ditentukan nilainya, seperti pada tabel 1 berikut:

Tabel 1. *Hyperparameter*

Parameter	Nilai
Batch Size	2
Epoch	25
Learning Rate	0.00003
Optimizer	SGD

Tabel 1. menampilkan konfigurasi *hyperparameter* yang digunakan dalam proses pelatihan model deteksi aktivitas merokok. *Batch size* ditetapkan sebesar 2, yang berarti model memproses dua gambar sekaligus dalam satu iterasi selama pelatihan, membantu dalam stabilisasi pembaruan bobot meskipun dengan sumber daya terbatas. Proses pelatihan dilakukan selama 25 *epoch*, di mana satu *epoch* merepresentasikan satu kali siklus penuh melalui seluruh dataset pelatihan, memberikan model kesempatan yang cukup untuk belajar dari data. *Learning rate* yang digunakan adalah sebesar 0.00003, nilai yang kecil ini bertujuan agar proses pembaruan bobot berlangsung secara perlahan dan stabil, sehingga dapat menghindari resiko *overshooting* atau tidak *konvergen*. Adapun algoritma optimisasi yang digunakan adalah *SGD (Stochastic Gradient Descent)*, yang merupakan optimizer paling umum digunakan untuk melatih model *machine learning* dan *deep learning*, terutama *neural network*. Dengan menggunakan parameter dengan nilai tersebut, diperoleh hasil pengujian sebagaimana dalam tabel 2 di bawah ini :

Tabel 1. Hasil pengujian *faster r-cnn*

No	Hasil	Pred	Act	Conf. Score	Threshold				
					0.5	0.6	0.7	0.8	0.9
1		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
2		Merokok	Merokok	0.97	TP	TP	TP	TP	TP

3		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
4		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
5		Merokok	Merokok	0.72	TP	TP	TP	FN	FN
6		Merokok	Merokok	0.85	TP	TP	TP	TP	FN
7		Merokok	Merokok	0.89	TP	TP	TP	TP	FN
8		Merokok	Merokok	0.82	TP	TP	TP	TP	FN
9		Merokok	Merokok	0.52	TP	FN	FN	FN	FN

10		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
11		Merokok	Merokok	0.51	TP	FN	TN	FN	FN
12		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
13		Merokok	Merokok	0.95	TP	TP	TP	TP	TP
14		Merokok	Merokok	0.92	TP	TP	TP	TP	TP
15		Merokok	Merokok	0.85	TP	TP	TP	TP	FN

16		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
17		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
18		Merokok	Merokok	0.76	TP	TP	TP	FN	FN
19		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
20		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
21		Merokok	Merokok	0.96	TP	TP	TP	TP	TP

22		Merokok	Merokok	0.82	TP	TP	TP	TP	FN
23		Merokok	Merokok	0.94	TP	TP	TP	TP	TP
24		Merokok	Merokok	0.75	TP	TP	TP	FN	FN
25		Merokok	Merokok	0.53	TP	FN	TN	TN	FN
26		Merokok	Merokok	0.94	TP	TP	TP	TP	TP
27		Merokok	Merokok	0.96	TP	TP	TP	TP	TP

28		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
29		Merokok	Merokok	0.94	TP	TP	TP	TP	TP
30		Merokok	Merokok	0.95	TP	TP	TP	TP	TP
31		Merokok	Merokok	0.93	TP	TP	TP	TP	TP
32		Merokok	Merokok	0.93	TP	TP	TP	TP	TP
33		Merokok	Merokok	0.79	TP	TP	TP	FN	FN

34		Merokok	Merokok	0.78	TP	TP	TP	FN	FN
35		Merokok	Merokok	0.78	TP	TP	TP	FN	FN
36		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
37		Merokok	Merokok	0.96	TP	TP	TP	TP	TP
38		Merokok	Merokok	0.98	TP	FP	TP	TP	TP
39		Merokok	Merokok	0.71	TP	TP	TP	FN	FN

40		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
----	---	---------	---------	------	----	----	----	----	----

Tabel 3. Hasil evaluasi *Faster r-cnn*

Nilai	Threshold				
	0.5	0.6	0.7	0.8	0.9
<i>True Positive</i>	40	36	37	30	25
<i>False Positive</i>	0	1	0	0	0
<i>True Negative</i>	0	0	0	0	0
<i>False Negative</i>	0	3	3	10	15
<i>Accuracy</i>	1	0.90	0.925	0.725	0.6
<i>Precision</i>	1	0.972	1	1	1
<i>Recall</i>	1	0.923	0.925	0.725	0.6
<i>F1-Score</i>	1	0.946	0.961	0.841	0.75

Adapun hasil dari Evaluasi model *faster r-cnn* pengujian yang dilakukan diperoleh hasil berikut:

1. Nilai *threshold* 0.5 membuat sebanyak 40 *true positive*, 0 *false positive*, 0 *true negative*, dan 0 *false negative*. Maka, diperoleh *accuracy* 100%, *precision* 100%, *recall* 100%, dan *f1-score* 100%.
2. Nilai *threshold* 0.6 membuat sebanyak 36 *true positive*, 1 *false positive*, 0 *true negative*, dan 3 *false negative*. Maka, diperoleh nilai *accuracy* 90%, *precision* 97,2%, *recall* 92,3%, dan *f1-score* 94,6%.
3. Nilai *threshold* 0.7 membuat sebanyak 37 *true positive*, 0 *false positive*, 0 *true negative*, dan 3 *false negative*. Maka, diperoleh nilai *accuracy* 92,5%, *precision* 100%, *recall* 92,5%, dan *f1-score* 96,1%.
4. Nilai *threshold* 0.8 membuat sebanyak 30 *true positive*, 0 *false positive*, 0 *true negative*, dan 10 *false negative*. Maka, diperoleh nilai *accuracy* 72,5%, *precision* 100%, *recall* 72,5%, dan *f1-score* 84,1%.
5. Nilai *threshold* 0.9 membuat sebanyak 25 *true positive*, 0 *false positive*, 0 *true negative*, 15 *false negative*. Maka, diperoleh nilai *accuracy* 60%, *precision* 100%, *recall* 60%, dan *f1-score* 75%.

Berdasarkan **Tabel 3.** hasil evaluasi performa model pada berbagai nilai *threshold* (0.5 hingga 0.9), dapat dilakukan analisis mendalam terhadap performa model dalam mendeteksi. Tabel ini mencakup metrik evaluasi seperti *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, *False Negative (FN)*, *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

Pada *threshold* 0.5, model menunjukkan performa sempurna dengan nilai *True Positive* sebanyak 40, tanpa adanya *False Positive* maupun *False Negative*, serta nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* semuanya bernilai 1.0. Ini menunjukkan bahwa pada *threshold* ini, model mampu mengidentifikasi seluruh data positif dengan sangat baik tanpa melakukan kesalahan prediksi. Namun demikian, tidak adanya *True Negative* maupun *False Positive* di semua *threshold* menunjukkan bahwa data uji mungkin hanya berisi sampel positif, atau model tidak pernah mendeteksi data sebagai negatif, sehingga analisis terhadap performa dalam membedakan kelas negatif tidak dapat dilakukan. Ketika *threshold* ditingkatkan menjadi 0.6, terlihat adanya penurunan *True Positive* menjadi 36 dan kemunculan 3 *False Negative*, yang menyebabkan *Recall* menurun menjadi 0.923. Meskipun demikian, *Precision* masih tinggi pada 0.972, dan *F1-Score* juga relatif tinggi di angka 0.946. Hal ini menunjukkan bahwa meskipun beberapa prediksi positif mulai terlewat (menjadi *FN*), sebagian besar prediksi positif masih benar (*TP*), dan secara keseluruhan performa model masih baik. Pada *threshold* 0.7, terdapat sedikit peningkatan pada *True Positive* (menjadi 37), namun *False Negative* tetap 3, menghasilkan *Recall* 0.925 dan *Precision* 1.0. Nilai *F1-Score* tertinggi dicapai di *threshold* ini, yaitu 0.961, yang menunjukkan bahwa keseimbangan antara *Precision* dan *Recall* sangat baik. Ini bisa menjadi indikasi bahwa *threshold* 0.7 adalah nilai optimal dalam konteks ini. Ketika *threshold* dinaikkan lebih tinggi ke 0.8 dan 0.9, performa model semakin menurun. Pada *threshold* 0.8, *True Positive* turun drastis menjadi 29, dan *False Negative* naik menjadi 11. Hal ini menyebabkan *Recall* turun ke 0.725 dan *F1-Score* ke 0.841, meskipun *Precision* masih tetap 1.0. Hal ini menunjukkan bahwa model menjadi terlalu ketat dalam menentukan prediksi positif, sehingga banyak data positif tidak dikenali (*FN* meningkat). Penurunan lebih lanjut terlihat pada *threshold* 0.9, dengan hanya 24 *True Positive* dan 16 *False Negative*, sehingga *Recall* menurun signifikan menjadi 0.6 dan *F1-Score* menjadi hanya 0.75. Secara keseluruhan, analisis ini menunjukkan bahwa nilai *threshold* sangat mempengaruhi performa model. *Threshold* rendah seperti 0.5 memberikan performa yang sangat tinggi pada dataset ini, sedangkan *threshold* tinggi menyebabkan model kehilangan banyak prediksi positif yang benar. Nilai *threshold* 0.7 dapat dianggap sebagai kompromi terbaik antara *Precision* dan *Recall*, sebagaimana terlihat dari nilai *F1-Score* yang tertinggi.

KESIMPULAN

Penelitian ini memfokuskan algoritma deteksi objek yang populer, yakni *Faster R-CNN*, untuk mengetahui mana yang lebih optimal dalam konteks tersebut. Dataset yang digunakan dalam penelitian ini diperoleh dari *Roboflow*, dengan total 1935 gambar beranotasi aktivitas merokok. Data ini kemudian dibagi menjadi data pelatihan dan validasi dengan rasio 85:15. Proses pelatihan model dilakukan di *Google Colab*, menggunakan *framework deep learning* berbasis *Python*. Setiap model diuji berdasarkan beberapa metrik evaluasi utama, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*, untuk memberikan gambaran yang komprehensif terhadap performa deteksi masing-masing algoritma. Hasil pelatihan menunjukkan bahwa model *Faster R-CNN* memiliki performa yang sangat baik. Pada nilai *threshold* 0.5, model ini mencapai skor sempurna di semua metrik evaluasi, yaitu 100% pada *accuracy*, *precision*, *recall*, dan *f1-score*. Bahkan ketika nilai *threshold* ditingkatkan menjadi 0.7, performanya masih tergolong sangat tinggi, dengan *f1-score* mencapai 96,1%. Ini mengindikasikan bahwa *Faster R-CNN* tidak hanya mampu mendeteksi objek rokok secara akurat, tetapi juga konsisten dalam berbagai skenario. Penurunan performa baru terlihat signifikan ketika *threshold* dinaikkan ke 0.8

atau 0.9, yang menandakan bahwa model menjadi terlalu selektif dan mulai kehilangan banyak deteksi positif. Ini menandakan bahwa Faster R-CNN sangat akurat dan konsisten, bahkan di berbagai skenario pengujian. Kemampuannya berasal dari arsitektur dua tahap—yang melibatkan *Region Proposal Network (RPN)* dan klasifikasi akhir, sehingga dapat mengidentifikasi objek secara lebih teliti. Meskipun waktu komputasi cenderung lebih lama, akurasi dan ketahanannya terhadap variasi *threshold* menjadikan *Faster R-CNN* sangat ideal untuk sistem pendeteksi otomatis dalam mendukung kawasan bebas rokok di kampus.

DAFTAR PUSTAKA

- Alamsyah, D., & Pratama, D. (2019). Deteksi Ujung Jari menggunakan Faster-RCNN dengan Arsitektur Inception v2 pada Citra Derau. *JuSiTik: Jurnal Sistem Dan Teknologi Informasi Komunikasi*, 2(1), 1. <https://doi.org/10.32524/jusitik.v2i1.435>
- Audina, M. T., Utaminingrum, F., & Syauqi, D. (2021). Sistem Deteksi dan Klasifikasi Jenis Kendaraan berbasis Citra dengan menggunakan Metode Faster-RCNN pada Raspberry Pi 4B. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 5(2), 814–819. <http://j-ptiik.ub.ac.id>
- Pardede, J., & Hardiansah, H. (2022). Deteksi Objek Kereta Api menggunakan Metode Faster R-CNN dengan Arsitektur VGG 16. *MIND Journal*, 7(1), 21–36. <https://doi.org/10.26760/mindjournal.v7i1.21-36>
- Petro, B. S. W., Mulyono, S., & Pramesti, V. A. (2024). *Prediksi Penyakit Batu Ginjal dengan Menerapkan Convolutional Neural Network*. 153–162.
- Pradana, I. C., Mulyanto, E., & Rachmadi, R. F. (2022). Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2. *Jurnal Teknik ITS*, 11(2). <https://doi.org/10.12962/j23373539.v11i2.86587>
- Putri, S. T. E., & Fahrurrozi, A. (2022). Pendeteksian Objek Pada Citra Hewan Karnivora Dan Herbivora Menggunakan Faster R-Cnn. *Jurnal Ilmiah Informatika Komputer*, 27(1), 32–42. <https://doi.org/10.35760/ik.2022.v27i1.5858>
- Ramadhanu, A., Ayu Mahessya, R., Raihan Zaky, M., Isra, M., Informasi, S., & Putra Indonesia YPTK Padang, U. (2023). Penerapan Teknologi Machine Learning Dengan Metode Vader Pada Aplikasi Sentimen Tamu Di Hotel Dymens. *JOISIE Journal Of Information System And Informatics Engineering*, 7(1), 165–173.
- Sudarto, S. (2020). Budaya Akademik Islami di Universitas Islam Sultan Agung Semarang dalam perspektif islamisasi ilmu. *Ta'dibuna: Jurnal Pendidikan Islam*, 9(2), 267. <https://doi.org/10.32832/tadibuna.v9i2.3526>
- Tivany Ramadhani, Usna Aulia, & Winda Amelia Putri. (2023). Bahaya Merokok Pada Remaja. *Jurnal Ilmiah Kedokteran Dan Kesehatan*, 3(1), 185–195. <https://doi.org/10.55606/klinik.v3i1.2285>