



REAL TIME OBJECT DETECTION MENGGUNAKAN FAST R-CNN

Salsabila Tsamrotul Qolbi¹⁾, Fitto Martcellindo²⁾, I Wayan Ardika Chandra³⁾, Naufal Adli⁴⁾, Ni Putu Dela Puspita⁵⁾, Abdul Haris⁶⁾

^{1,2,3,4,6} Program Studi Teknik Informatika, Institut Teknologi PLN

⁵ Program Studi Teknik Sipil, Institut Teknologi PLN

^{1,2,3,4,5,6} Kota Cengkareng, Jakarta Barat

Email: ¹salsabilachasa@gmail.com, ²fitto2131001@itpln.ac.id, ³iwayanchandra2003@gmail.com,

⁴naufal2131006@itpln.ac.id, ⁵puspitadela131@gmail.com, ⁶harismwakang@itpln.ac.id

Abstract

Intelligent computing technology has become the main foundation for various innovations, including the development of object detection systems. In this research, an object detection system based on the Fast Region-based Convolutional Neural Network (Fast R-CNN) algorithm was designed. This method was chosen due to its superior ability in detection speed and accuracy compared to previous algorithms. By integrating feature extraction and classification processes directly, the Fast R-CNN method is able to produce more efficient detection. Testing was carried out using object datasets with variations in lighting, angles, and object sizes. The results show that this method is able to detect objects with a high level of accuracy. The processing speed of Fast R-CNN makes it suitable for real-time dataset testing in applications such as security, surveillance, and autonomous systems. With this approach, it is hoped that this research can make a real contribution to the development of more reliable and effective object detection technology in various areas of life.

Keyword: Intelligent Computing Technology, Object Detection, Fast R-CNN.

Abstrak

Teknologi komputasi cerdas telah menjadi dasar utama dalam berbagai inovasi, termasuk pengembangan sistem deteksi objek. Dalam penelitian ini, dirancang sebuah sistem deteksi objek berbasis algoritma *Fast Region-based Convolutional Neural Network* (Fast R-CNN). Metode ini dipilih karena kemampuannya yang unggul dalam kecepatan dan akurasi deteksi dibandingkan dengan algoritma sebelumnya. Dengan mengintegrasikan proses ekstraksi fitur dan klasifikasi secara langsung, metode Fast R-CNN mampu menghasilkan deteksi yang lebih efisien. Pengujian dilakukan menggunakan dataset objek dengan variasi pencahayaan, sudut pandang, dan ukuran objek. Hasilnya menunjukkan bahwa metode ini mampu mendeteksi objek dengan tingkat akurasi yang tinggi. Kecepatan pemrosesan yang dimiliki Fast R-CNN menjadikannya cocok untuk pengujian dataset secara real-time dalam aplikasi seperti keamanan, pengawasan, dan sistem otonom. Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan teknologi deteksi objek yang lebih andal dan efektif di berbagai bidang kehidupan.

Kata Kunci: Teknologi Komputasi Cerdas, Deteksi Objek, Fast R-CNN.

1. PENDAHULUAN

Deteksi objek adalah salah satu teknik utama dalam visi komputer yang digunakan untuk mengidentifikasi dan menentukan lokasi objek dalam gambar atau video [1]. Teknik ini menjadi fondasi berbagai aplikasi canggih, seperti pengenalan pola, sistem keamanan, dan pemantauan cerdas. Algoritma deteksi objek modern biasanya mengintegrasikan machine learning atau deep learning untuk menghasilkan deteksi yang presisi dan akurat [2]. Deteksi wajah, khususnya, telah berkembang menjadi teknologi yang sangat signifikan dalam bidang keamanan, autentikasi biometrik, serta interaksi manusia-komputer. Deteksi wajah berfungsi sebagai langkah awal yang krusial dalam proses identifikasi wajah, yang bertujuan untuk menghasilkan akurasi yang optimal [3].

Wajah memiliki karakteristik unik dan kompleks yang menjadikannya elemen penting dalam berbagai aplikasi, termasuk sistem keamanan dan absensi berbasis biometrik. Salah satu algoritma unggulan untuk deteksi objek adalah Fast R-CNN, yang merupakan algoritma berbasis *Convolutional Neural Network* (CNN) dengan kemampuan unggul dalam mendeteksi objek secara efisien [4]. Deteksi wajah merupakan elemen mendasar dalam aplikasi seperti *face recognition*, analisis ekspresi wajah, dan pengenalan emosi. Tujuan utamanya adalah mendeteksi keberadaan wajah dalam sebuah video atau gambar, serta menentukan posisinya secara akurat. Fast R-CNN memiliki tiga modul utama: pertama, modul *deep fully convolutional network* yang mengusulkan *region*; kedua, modul detector Fast R-CNN yang menggunakan



region yang diusulkan dari RPN (*Region Proposal Network*); dan ketiga, modul untuk melakukan klasifikasi berdasarkan data yang dihasilkan [5].

Pendekatan yang digunakan dalam deteksi wajah mencakup dua jenis utama, yaitu *multi-stage detector* dan *one-stage detector*, yang menggabungkan proses usulan wilayah (*region proposal*) dan klasifikasi secara menyeluruh [6]. Algoritma yang digunakan untuk deteksi objek ini memanfaatkan kemampuan pembelajaran mesin yang terus berkembang untuk menghasilkan hasil yang lebih baik dan konsisten [7].

Dari beberapa study literatur yang telah dilakukan seperti pada jurnal Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video dapat diimplementasikan bahwa dari dataset yang berupa foto dan video klip yang telah di rekam dan diolah menggunakan algoritma Fast R-CNN dengan bahasa pemrograman berupa python, tensorflow dan openCV menghasilkan nilai akurasi rata-rata sebesar 97,07% pada tahap training dengan 21 epoch. Dapat disimpulkan bahwa implementasi algoritma Fast R-CNN mampu membedakan antara wajah spoof dan non spoof dengan tingkat akurasi yang tinggi, yang dapat mendukung keamanan sistem otentikasi berbasis analisa wajah. Pada penelitian lain yang dikutip pada jurnal dengan judul Analisa Penggunaan Metode Faster R-CNN dalam Pengenalan Wajah: Systematic Literature Review dapat diimplementasikan bahwa algoritma Fast R-CNN menggabungkan Region Proposal Network dan Convolutional Neural Network untuk mempercepat proses deteksi objek tanpa mengurangi akurasi. Dengan demikian, implementasi metode Fast R-CNN dalam proses deteksi wajah menunjukkan efektivitas persentasi akurasi yang tinggi dan meminimalisir kesalahan dalam deteksi wajah.

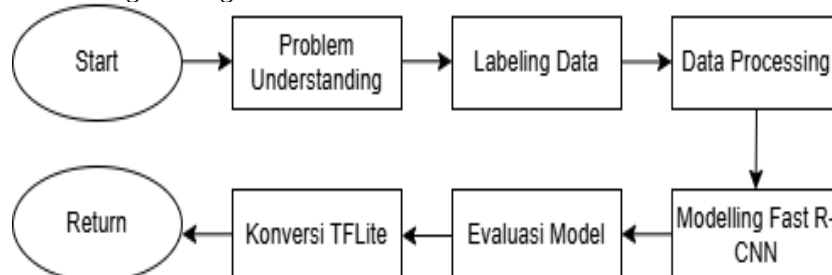
Meskipun berbagai penelitian telah dilakukan, masih terdapat tantangan (*gap*) yang perlu diatasi, yaitu bagaimana mengembangkan sistem deteksi wajah yang tidak hanya memiliki akurasi tinggi tetapi juga performa yang cepat untuk mendukung aplikasi *real-time*. Penelitian ini bertujuan untuk menjawab tantangan tersebut dengan memanfaatkan algoritma Fast R-CNN yang dioptimalkan. Tujuan utama penelitian ini adalah mengembangkan sistem deteksi wajah berbasis Fast R-CNN yang mengutamakan kecepatan dan akurasi secara bersamaan. Selain itu, penelitian ini diharapkan memberikan kontribusi signifikan dalam kemajuan teknologi deteksi wajah yang lebih adaptif dan relevan untuk masa depan.

Dengan pemanfaatan Fast R-CNN yang dioptimalkan, penelitian ini berpotensi besar untuk mengatasi kendala utama dalam pengembangan teknologi deteksi wajah. Fokusnya adalah pada peningkatan kecepatan dan akurasi sistem, sehingga hasil penelitian ini dapat diterapkan dalam berbagai sektor, mulai dari sistem absensi berbasis biometrik hingga interaksi manusia-komputer yang lebih efisien. Dengan demikian, penelitian ini diharapkan memberikan dampak positif yang luas dalam berbagai aspek kehidupan, sekaligus memperkuat fondasi teknologi deteksi wajah di masa mendatang.

ujuan dari penelitian ini adalah untuk mengukur tingkat akurasi metode Fast R-CNN dalam sistem absensi berbasis kamera, guna memastikan keandalan dan ketepatan dalam mendeteksi wajah pengguna. Selain itu, penelitian ini juga bertujuan untuk menganalisis efisiensi penggunaan metode Fast R-CNN dalam proses pendeteksian wajah, sehingga dapat diketahui sejauh mana metode ini mampu mengidentifikasi individu dengan cepat dan akurat. Selain aspek teknis, penelitian ini juga mengevaluasi tingkat keamanan serta perlindungan data privasi pengguna dalam sistem absensi berbasis kamera, guna memastikan bahwa informasi pribadi tetap terjaga dan tidak disalahgunakan.

2. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan utama yang dirancang untuk mengembangkan dan menguji aplikasi deteksi wajah dan objek berbasis metode Fast R-CNN. Untuk memastikan proses penelitian berjalan dengan menghasilkan aplikasi yang optimal, diterapkan beberapa tahapan metode penelitian. Tahapan tersebut meliputi pemahaman masalah, pelabelan data, pemrosesan data, pemodelan, dan evaluasi sistem. setelah model berhasil dikembangkan, model dikonversi ke format TensorFlow Lite agar dapat diimplementasikan pada aplikasi. Berikut adalah penjelasan lebih rinci terkait langkah-langkah tersebut.



Gambar 1 metode penelitian



Gambar 1 menunjukkan alur metode penelitian yang terdiri dari beberapa tahap utama untuk mencapai hasil akhir. Penjelasan dari masing-masing tahap adalah sebagai berikut:

2.1 Problem Understanding

Tahap ini merupakan langkah awal yang krusial dalam proses pengembangan sistem deteksi objek berbasis Fast R-CNN. Pada tahap ini, dilakukan analisis mendalam terhadap esensi masalah yang akan diselesaikan, termasuk identifikasi kebutuhan utama dan tujuan spesifik dari sistem deteksi objek. Fokus utama adalah memahami karakteristik dataset yang akan digunakan, seperti variasi pencahayaan, sudut pandang, dan ukuran objek, serta mengkaji tantangan yang berpotensi memengaruhi akurasi dan efisiensi deteksi. Dengan memahami konteks masalah ini, strategi pengembangan sistem dapat dirancang secara optimal untuk mencapai tingkat akurasi dan kecepatan yang diharapkan dalam berbagai aplikasi, seperti pengawasan, keamanan, dan sistem otonom. Pada permasalahan kali ini diberikan data dengan total jumlah 71 data citra dengan __data citra wajah Wayan, __data citra wajah Fitto, __data citra wajah Adli, __data citra wajah Dela, __data citra wajah Haris, data citra wajah Caca dan data citra wajah Ibnu. Dengan contoh data pada table 1 berikut

Table 1 *LabelImg*

Fitto	Wayan	Adli	Caca	Dela	Ibnu	Haris
						

A. Proses Analisis

1. Pemahaman Dataset:

- o Menganalisis total data citra sebanyak 71, serta mendistribusikannya ke masing-masing kategori wajah. Identifikasi jumlah citra untuk setiap individu perlu dilakukan untuk mengamati sebaran data dan potensi ketidakseimbangan data.
- o Memahami karakteristik dataset seperti variasi pencahayaan, sudut pengambilan gambar, resolusi gambar, dan kualitas citra.

2. Preprocessing Dataset:

- o Memastikan semua citra memiliki ukuran dan format konsisten untuk diproses oleh model Fast R-CNN.
- o Melakukan augmentasi data seperti rotasi, flip, dan perubahan pencahayaan untuk meningkatkan keberagaman dataset.

3. Labeling Data:

- o Memberikan label yang sesuai untuk setiap citra berdasarkan kategori wajah (Fitto, Wayan, Adli, Caca, Dela, Ibnu, Haris).

4. Eksplorasi dan Analisis Data:

- o Menghitung distribusi jumlah citra per kategori wajah untuk mengidentifikasi potensi ketidakseimbangan data.
- o Melakukan visualisasi data untuk memahami pola dan variasi dalam dataset.

5. Penerapan Model Fast R-CNN:

- o Menggunakan dataset terlabel untuk melatih model Fast R-CNN. Model ini akan mendeteksi dan mengklasifikasikan wajah berdasarkan kategori yang telah ditentukan.
- o Mengevaluasi performa model dengan metrik seperti akurasi, precision, recall, dan F1-score.

B. Hasil Analisis

1. Distribusi Data:

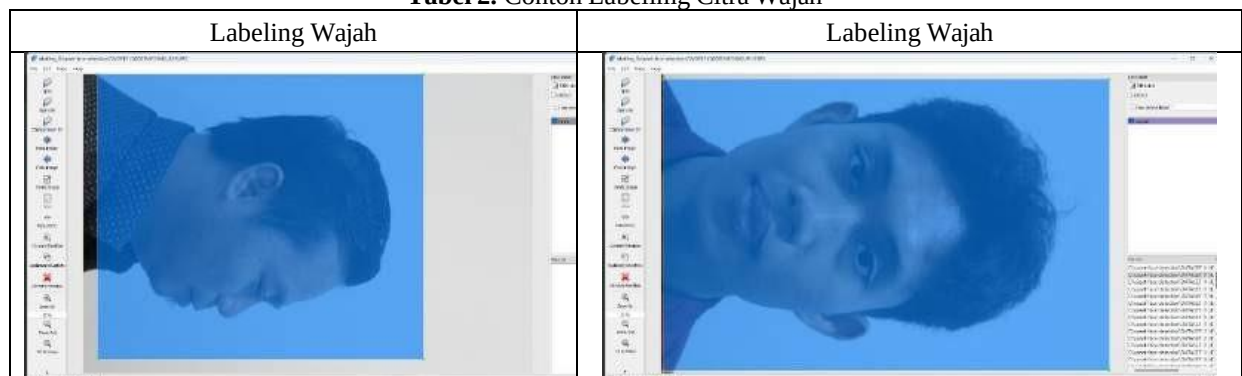
- o Menyediakan tabel atau grafik distribusi jumlah data citra untuk setiap kategori wajah. Misalnya:
 - Wayan: __data citra
 - Fitto: __data citra

- Adli: __data citra
 - Dela: __data citra
 - Haris: __data citra
 - Caca: __data citra
 - Ibnu: __data citra
2. Temuan Karakteristik Dataset:
 - Variasi pencahayaan dan sudut pandang dapat memengaruhi performa deteksi.
 - Ketidakseimbangan data (misalnya, kategori dengan lebih banyak data dapat memengaruhi bias model).
 3. Evaluasi Model:
 - Setelah melatih model Fast R-CNN, dapat disajikan hasil evaluasi seperti berikut:
 - Akurasi: __
 - Precision per kategori: Wayan (%), Fitto (%), dst.
 - Recall per kategori: Wayan (%), Fitto (%), dst.
 - F1-Score: __
 - Hasil evaluasi menunjukkan bahwa model lebih akurat mendeteksi wajah dengan jumlah data yang lebih banyak, sementara akurasi untuk kategori dengan data lebih sedikit lebih rendah.

2.2 Labeling Data

Tahap Labeling Data merupakan langkah penting dalam proses pengembangan sistem deteksi objek berbasis Fast R-CNN. Labeling data merupakan proses untuk mengidentifikasi data mentah untuk memberikan konteks [8]. Pada langkah ini, dilakukan pelabelan terhadap dataset yang akan digunakan untuk pelatihan model. Proses pelabelan mencakup identifikasi dan anotasi objek-objek dalam gambar sesuai dengan kategori yang telah ditentukan. Dataset yang digunakan dilengkapi dengan variasi kondisi, seperti pencahayaan, sudut pandang, dan ukuran objek, sehingga memastikan data yang dihasilkan memiliki representasi yang baik terhadap situasi nyata. Pelabelan yang akurat berperan penting dalam meningkatkan kemampuan model untuk mendeteksi objek dengan tingkat akurasi tinggi, sekaligus meminimalkan kesalahan dalam prediksi selama proses pengujian dan implementasi di aplikasi nyata.

Tabel 2. Contoh Labeling Citra Wajah



Berikut adalah tabel yang merangkum langkah-langkah proses labeling data untuk pengembangan sistem deteksi objek berbasis Fast R-CNN.

Table 2 Deskripsi LabelImg

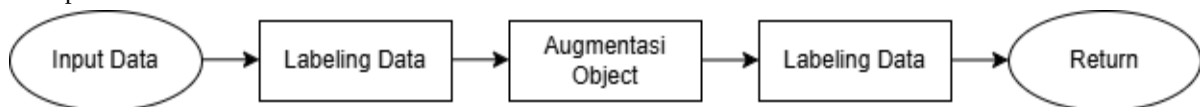
No	Langkah	Deskripsi
1	Persiapan Dataset	- Kumpulkan citra dari setiap kategori (Wayan, Fitto, Adli, Caca, Dela, Ibnu, Haris). - Organisasi dataset dalam folder terpisah untuk setiap kategori wajah.
2	Alat Labeling	- LabelImg yang mendukung format Pascal VOC atau COCO.
3	Proses Labeling Data	- Impor dataset ke alat labeling. - Gunakan bounding box untuk menandai wajah pada gambar dan berikan label sesuai kategori (misalnya: Wayan).
4	Augmentasi Data	- Verifikasi semua bounding box dan label untuk memastikan anotasi akurat. - Tingkatkan variasi data dengan rotasi, flipping, zoom, atau perubahan pencahayaan



5	Ekspor Dataset	- Simpan hasil augmentasi dalam folder yang sesuai. - Ekspor dataset dalam format yang kompatibel dengan Fast R-CNN (XML untuk Pascal VOC atau JSON untuk COCO).
6	Evaluasi Dataset	- Organisasi file ekspor agar sesuai dengan struktur folder dataset. - Periksa distribusi data per kategori wajah untuk memastikan keseimbangan data.
7	Persiapan Dataset	- Verifikasi kembali semua label untuk menghindari kesalahan. - Bagi dataset menjadi training set (80%), validation set (10%), dan testing set (10%). - Organisasi dataset dalam folder sesuai kebutuhan model Fast R-CNN.

2.3 Data Preprocessing

Data preprocessing merupakan tahapan untuk menghilangkan beberapa permasalahan yang bisa mengganggu saat pemrosesan data [9]. Tahap ini sering disebut sebagai tahap yang padat karya. Aktivitas yang dilakukan antara lain pengumpulan atau “*importing*” data, Augmentasi data, dan pemisahan data. alur pengerjaan *data preprocessing* ini dapat dilihat pada Gambar 2.



Gambar 2 Data Preprocessing

Tahap *Data Preprocessing* merupakan proses memanipulasi data untuk menghasilkan data yang lebih berarti digunakan untuk pelatihan model Fast R-CNN [10]. Proses ini mencakup serangkaian langkah untuk memastikan data yang digunakan memiliki kualitas yang optimal dan konsisten. Dalam konteks penelitian ini, *preprocessing* melibatkan normalisasi gambar untuk mengatasi variasi pencahayaan, penyelarasan ukuran gambar agar seragam, serta penghilangan noise yang dapat mengganggu proses pelatihan. Selain itu, dilakukan augmentasi data seperti rotasi, flipping, atau perubahan skala untuk memperluas variasi dataset dan meningkatkan kemampuan generalisasi model. Dengan *preprocessing* yang baik, dataset menjadi lebih representatif terhadap kondisi nyata, sehingga model dapat mencapai akurasi deteksi yang lebih tinggi dalam berbagai situasi.

2.4 Modelling

Langkah menetapkan teknik *deep learning* yang akan digunakan melibatkan pemilihan *tools*, teknik, algoritma, dan penentuan *hyperparameter* yang paling sesuai untuk mencapai tingkat akurasi optimal. Pendekatan yang diterapkan adalah metode klasifikasi dalam *deep learning*, khususnya menggunakan algoritma CNN yang di modifikasi agar dapat menjadi sebuah model yang bertipe *Object-detection*. *Deep learning* adalah metode *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan [11].

Prosedur ini menjelaskan tahapan dalam pengaturan dan pelatihan model untuk deteksi objek dengan menggunakan kerangka kerja TensorFlow *Object Detection*. Pertama, parameter untuk pelatihan seperti total langkah dan ukuran batch ditetapkan. Kemudian, lokasi berkas konfigurasi model serta model yang akan digunakan untuk pelatihan ulang ditentukan. Sebuah fungsi digunakan untuk menghitung total kelas dalam dataset berdasarkan berkas pemetaan label, yang selanjutnya diterapkan dalam pengaturan model.

Tahap berikutnya mencakup pembuatan berkas konfigurasi kustom dengan menyesuaikan berbagai elemen penting, termasuk jalur berkas untuk dataset pelatihan dan pengujian, jalur pemetaan label, ukuran batch, total langkah pelatihan, dan jumlah kelas yang telah dihitung sebelumnya. Di samping itu, beberapa pengaturan khusus diterapkan sesuai dengan model yang dipilih, contohnya penyesuaian laju pembelajaran untuk model tertentu atau alternatif dalam cara perubahan ukuran gambar untuk model yang lain.

Setelah berkas konfigurasi telah disesuaikan, model dan data input dipersiapkan dengan menggunakan pengaturan yang telah dimodifikasi. Pelatihan model pun dimulai dengan menerapkan strategi pelatihan paralel, yang memungkinkan penggunaan beberapa perangkat untuk mempercepat proses pelatihan. Selama proses pelatihan, model disimpan dalam folder yang telah ditetapkan dengan titik pemulihan yang disimpan setiap beberapa langkah. Proses ini memungkinkan model deteksi objek dilatih dengan baik untuk mengenali berbagai kategori objek dalam dataset yang tersedia. Berikut ini beberapa tahapan (*step by step*) pemodelan data:



Table 2 Deskripsi Modelling

No	Tahapan	Deskripsi
1	Pemilihan Teknik dan Algoritma	- Pilih algoritma CNN yang dimodifikasi untuk deteksi - Gunakan TensorFlow Object Detection API sebagai - Tentukan hyperparameter seperti learning rate, batch size, epoch, dan augmentasi data.
2	Persiapan Dataset	- Kumpulkan dataset wajah (contoh: Wayan, Fitto, Adli, Caca, Dela, Ibnu, Haris). - Konversi dataset ke format TFRecord untuk kompatibilitas dengan TensorFlow. - Buat file label_map.pbtxt untuk memetakan label ke ID kategori. - Pilih model pra-latih (contoh: Faster R-CNN, SSD, atau EfficientDet) dari TensorFlow Model Zoo.
3	Pembuatan Berkas Konfigurasi	- Atur jalur dataset pelatihan/pengujian, label map, batch size, dan langkah pelatihan pada file konfigurasi. - Sesuaikan hyperparameter seperti learning rate dan augmentasi data.
4	Pelatihan Model	- Jalankan pelatihan menggunakan perintah dari TensorFlow Object Detection API. - Gunakan TensorBoard untuk memantau metrik seperti loss, precision, dan recall. - Simpan checkpoints model secara berkala selama proses pelatihan.
5	Evaluasi Model	- Gunakan dataset pengujian untuk mengevaluasi model. - Hitung metrik seperti mAP, recall, dan F1-Score untuk menilai kinerja model.
6	Fine-tuning dan Optimasi	- Analisis hasil pelatihan untuk mengidentifikasi area yang memerlukan optimasi. - Ubah hyperparameter seperti learning rate atau augmentasi data, kemudian lakukan pelatihan ulang.
7	Implementasi Model	- Ekspor model terlatih ke format saved_model untuk digunakan dalam aplikasi. - Integrasikan model ke aplikasi dan lakukan pengujian akhir dengan data nyata.

2.5 Evaluasi

Pada tahap evaluasi dilaksanakan secara menyeluruh dengan tujuan menyelaraskan model yang dihasilkan agar sesuai dengan tujuan yang telah ditetapkan pada tahap awal. Hasil implementasi dari evaluasi dalam penelitian ini memanfaatkan mAP (*mean Average Precision*) yang merupakan rata-rata presisi untuk mengukur kinerja model deteksi objek secara keseluruhan. Rumus dasar untuk menghitung AP (*Average Precision*) adalah

$$AP = \frac{1}{n} \sum (R_n - R_{n-1}) P_n$$

di mana:

- P_n = Presisi pada titik ke- n
- R_n = Recall pada titik ke- n

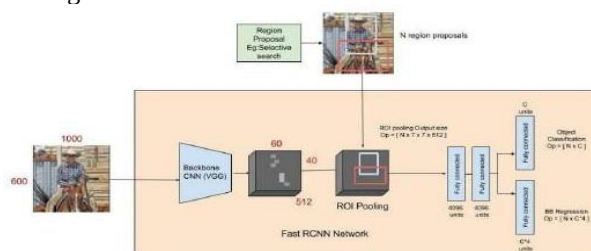
Model object detection dalam evaluasi ini menghasilkan dua jenis output utama yaitu :

- Kelas prediksi (*class Prediction*) yaitu mengidentifikasi jenis objek yang terdeteksi.
- Bounding box regression* yaitu menghasilkan koordinat kotak pembatas yang menunjukkan lokasi objek dalam gambar.

3. HASIL DAN PEMBAHASAN

3.1 Arsitektur CNN

Fast R-CNN (*Region-based Convolutional Neural Network*) merupakan salah satu arsitektur dalam bidang deteksi objek yang dikembangkan untuk meningkatkan kecepatan dan akurasi dalam mengenali serta mengklasifikasikan objek pada suatu gambar.



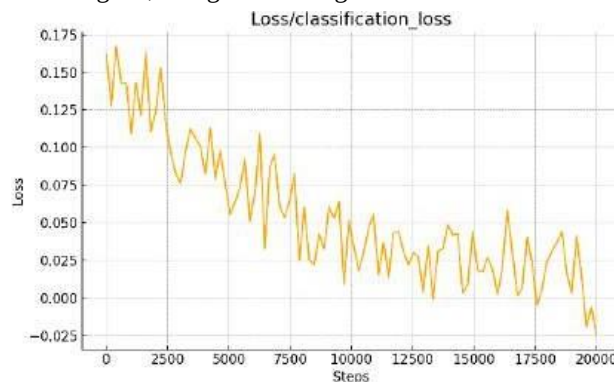
Gambar 3 Arsitektur Fast R-CNN



Fast R-CNN merupakan metode deteksi objek yang dirancang untuk meningkatkan efisiensi dan kecepatan dibandingkan pendahulunya, R-CNN. Algoritma Fast R-CNN diklaim mampu mengurangi waktu training dalam jumlah besar [12]. Arsitektur ini terdiri dari beberapa tahapan utama sebagai berikut:

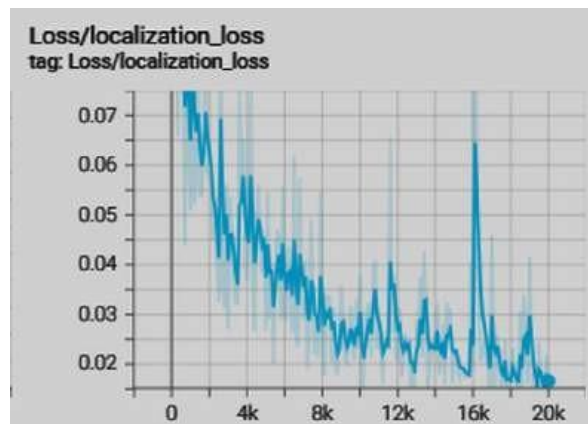
1. **Input Gambar**
Pada tahap ini, gambar input dengan resolusi tetap, misalnya berukuran 600×1000 , diberikan sebagai masukan ke model. Gambar ini digunakan untuk mendeteksi objek yang ada di dalamnya. Pemilihan resolusi ini tetap membantu model dalam menormalkan dimensi gambar sehingga kompatibel dengan jaringan yang digunakan.
2. **Ekstraksi Fitur dengan CNN**
Jaringan Convolutional Neural Network (CNN), seperti arsitektur VGG, digunakan sebagai backbone untuk mengekstraksi fitur dari gambar. Proses ini menghasilkan feature map dengan dimensi $H \times W \times CH \times W \times CH \times W \times C$, di mana HHH dan WWW adalah dimensi spasial (contoh: 60×40), dan CCC adalah jumlah saluran fitur contohnya 512. Peta fitur ini menjadi representasi penting dari gambar input karena berisi informasi visual yang diperlukan untuk mendeteksi objek.
3. **Region Proposal**
Metode seperti Selective Search digunakan untuk menghasilkan region proposals (area kandidat yang kemungkinan besar berisi objek). Region proposals ini kemudian dipetakan ke feature map menggunakan koordinat spasial.
4. **ROI Pooling (Region of Interest Pooling)**
Untuk setiap region proposal, dilakukan ROI pooling untuk menyesuaikan ukuran fitur menjadi dimensi tetap, misalnya $7 \times 7 \times 512 \times 7 \times 512$. Proses ini memastikan bahwa semua proposal dapat diproses secara seragam, terlepas dari ukuran aslinya. ROI pooling juga membantu mengurangi dimensi data tanpa mengorbankan informasi penting.
5. **Lapisan Fully Connected (FC)**
Fitur hasil ROI pooling diteruskan ke dua lapisan fully connected (FC) yang masing-masing terdiri dari 4096 unit. Tahapan ini bertujuan untuk mengekstraksi informasi tingkat tinggi dari fitur.
6. **Cabang untuk Klasifikasi dan Regresi**
Setelah melalui lapisan FC, jaringan bercabang menjadi dua jalur terpisah:
 - a. **Object Classification:** Cabang pertama bertanggung jawab untuk mengklasifikasikan objek pada setiap region proposal. Output berupa probabilitas untuk setiap kelas, dengan ukuran $[N \times C][N \times C]$, di mana NNN adalah jumlah region proposals, dan CCC adalah jumlah kelas objek.
 - b. **Bounding Box Regression:** Cabang kedua bertugas memperbaiki posisi dan ukuran bounding box untuk setiap proposal. Outputnya berupa regresi untuk empat parameter bounding box (x, y, w, h), dengan ukuran $[N \times 4][N \times 4]$.

Di latih sebanyak 20.000 langkah, menghasilkan 3 grafik loss

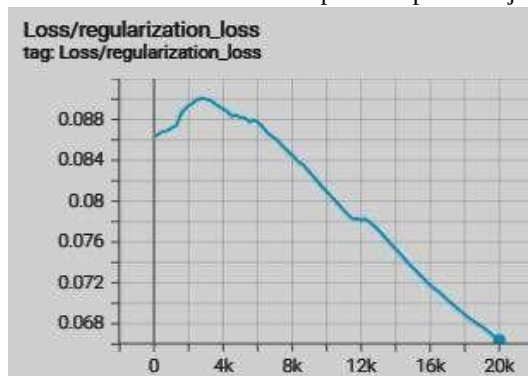


Gambar 4 Grafik Loss

Plot ini menggambarkan perubahan nilai *classification loss* selama pelatihan. Nilai *classification loss* digunakan untuk mengukur seberapa baik model dalam memprediksi kelas yang benar untuk setiap sampel. Penurunan nilai ini secara konsisten menunjukkan bahwa model semakin baik dalam melakukan klasifikasi data yang tersedia.

**Gambar 5** Classification Loss

Gambar 5 menggambarkan *localization loss* yaitu metrik yang digunakan dalam model deteksi objek untuk mengatur seberapa akurat prediksi lokasi objek, terutama dalam menentukan posisi *bounding box* yang sesuai. *Localization loss* mengukur kesalahan antara posisi *bounding box* yang diprediksi oleh model dengan posisi *bounding box* sebenarnya pada data pelatihan. Nilai *localization loss* lebih kecil menunjukkan bahwa model semakin baik dalam mengenali posisi objek secara akurat. Penurunan nilai *localization loss* menunjukkan bahwa model semakin baik dalam memprediksi posisi objek pada data latih.

**Gambar 6** Localisation Loss

Regularization loss adalah penalti yang diterapkan pada bobot model untuk mengurangi risiko *overfitting*. Plot ini menunjukkan bahwa nilai *regularization loss* menurun selama pelatihan, yang merupakan indikasi bahwa model mengontrol kompleksitasnya dengan baik.

Disini bisa kita lihat pada setiap loss, model dapat belajar dengan baik, karena siringinya langkah dalam pelatihan nilai loss masing-masing semakin mengecil, meskipun harus naik atau turun seperti pada *Localisation* dan *Classification loss*, namun hasil akhir dari pelatihan model ini adalah dari ketiganya loss nya kurang dari 0.1.

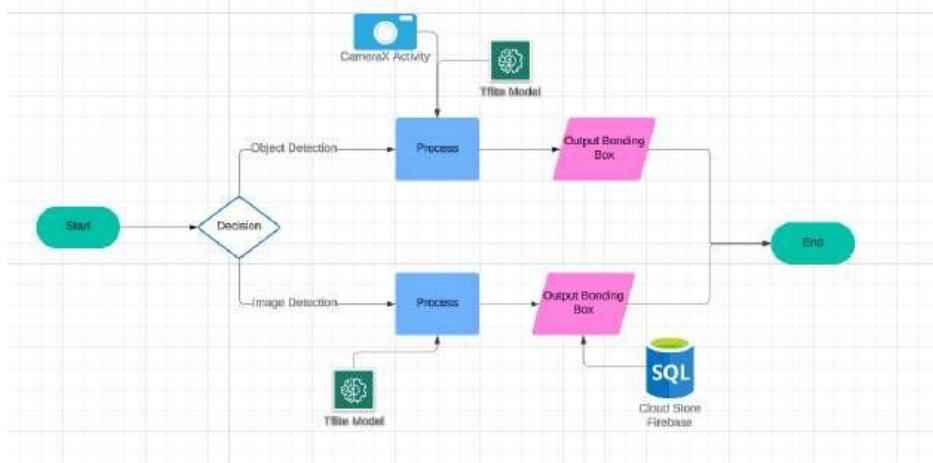
3.2 Konversi model TF Lite

Model yang sudah dilatih awalnya disimpan dalam format hdf5, namun sesuai dengan kebutuhan perangkat mobile, maka model tersebut haruslah di konversi menjadi format tflite yang bertujuan agar model tersebut dapat digunakan pada perangkat daya rendah atau perangkat mobile. Proses konversi ini dilakukan setelah model disimpan dalam format hdf5, dengan tahapan awal membaca ulang model tersebut lalu mendeklarasikan optimasi yang akan digunakan pada model. Hasilnya model dengan format hdf5 berukuran sekitar 450 Mb dapat dikonversi menjadi 54 Mb.

3.3 Implementasi Model

Gambar berikut merupakan diagram alur yang menggambarkan proses implementasi sistem deteksi berbasis kecerdasan buatan (AI) menggunakan model TensorFlow Lite (TFLite). Diagram ini menunjukkan bagaimana data diolah mulai dari tahap awal hingga menghasilkan output yang dapat digunakan lebih lanjut. Selain itu, sistem juga terintegrasi

dengan teknologi penyimpanan berbasis cloud seperti SQL, Cloud Store, dan Firebase, yang memungkinkan data hasil deteksi tersimpan dengan aman untuk keperluan analisis lebih lanjut. Diagram ini memberikan gambaran umum tentang bagaimana teknologi deteksi gambar dan objek dapat diterapkan dalam berbagai sistem berbasis AI.



Gambar 7 Implementasi Model

Pada Gambar 7 diatas menunjukkan alur kerja aplikasi yang menggunakan analisis gambar berbasis FAST R CNN dan penyimpanan data di Firestore dan SQL Server. Proses dimulai dari elemen hijau bertuliskan "Start", yang berfungsi sebagai pemicu untuk mengaktifkan sistem. Setelah itu, sistem memasuki tahap Decision, di mana keputusan dibuat untuk menentukan apakah proses yang akan dijalankan adalah Object Detection atau Image Detection. Keputusan ini didasarkan pada input atau parameter awal yang diberikan oleh pengguna atau sistem. Berdasarkan hasil keputusan, alur terbagi menjadi dua cabang utama, yaitu Object Detection dan Image Detection. Pada jalur Object Detection, sistem menggunakan CameraX Activity untuk menangkap gambar atau video dari objek yang akan dianalisis. Kamera ini terintegrasi dengan framework CameraX, yang banyak digunakan dalam aplikasi Android untuk pengambilan gambar. Setelah gambar diperoleh, data tersebut dikirimkan ke TfLite Model (TensorFlow Lite), yang bertugas memproses gambar dan mendeteksi objek di dalamnya. Selanjutnya, tahap Process dilakukan untuk menganalisis fitur gambar menggunakan algoritma deteksi berbasis TfLite. Hasil dari proses ini berupa Output Bounding Box, yaitu kotak pembatas yang menunjukkan lokasi objek yang terdeteksi dalam gambar. Bounding box ini kemudian disiapkan sebagai output untuk ditampilkan kepada pengguna, dan setelah itu, sistem menyelesaikan proses dan memasuki tahap End. Sementara itu, pada jalur Image Detection, gambar dari input diproses menggunakan TfLite Model untuk mendeteksi fitur atau pola spesifik dalam gambar. Proses ini bertujuan untuk mengidentifikasi objek berdasarkan tugas yang telah ditentukan, seperti pengenalan wajah atau klasifikasi gambar. Hasil dari deteksi ini juga berupa Output Bounding Box atau informasi lainnya, seperti label atau kategori objek yang dikenali. Selanjutnya, hasil deteksi ini disimpan dalam sistem berbasis cloud, seperti Firebase Cloud Firestore atau SQL, agar dapat dianalisis lebih lanjut atau digunakan kembali di masa mendatang. Setelah data berhasil disimpan atau bounding box dihasilkan, sistem menyelesaikan proses dan masuk ke tahap End.

Dengan mengintegrasikan proses ekstraksi fitur dan klasifikasi secara langsung, metode Fast R-CNN mampu menghasilkan deteksi yang lebih efisien. Pengujian dilakukan menggunakan dataset objek dengan variasi pencahayaan, sudut pandang, dan ukuran objek. Hasilnya menunjukkan bahwa metode ini mampu mendeteksi objek dengan tingkat akurasi yang tinggi. Kecepatan pemrosesan yang dimiliki Fast R-CNN menjadikannya cocok untuk pengujian dataset secara real-time dalam aplikasi seperti keamanan, pengawasan, dan sistem otonom. Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan teknologi deteksi objek yang lebih andal dan efektif di berbagai bidang kehidupan.

Table 4 Hasil Pengujian

No	Fitur	Hasil Fitur	Keterangan
1	Realtime Object	Dapat menampilkan hasil Bounding box per obejct.	Sesuai
2	Input Gambar	Dapat memasukkan gambar	Sesuai
3	Deteksi Gambar	Dapat mendektesi objek dengan baik	Sesuai

3.4 Aplikasi Android Objek Detection

3.4.1 Model Deploy

Pada tahap ini model di export dengan format flat buffer Tensorflow Lite yakni (.tflite). Dalam Tensorflow Lite versi 0.4.4 metadata, spesifikasi model AverageWordVec secara default melakukan eksport vocab file dan label menjadi metadata model.

3.4.2 Activity Menu Utama

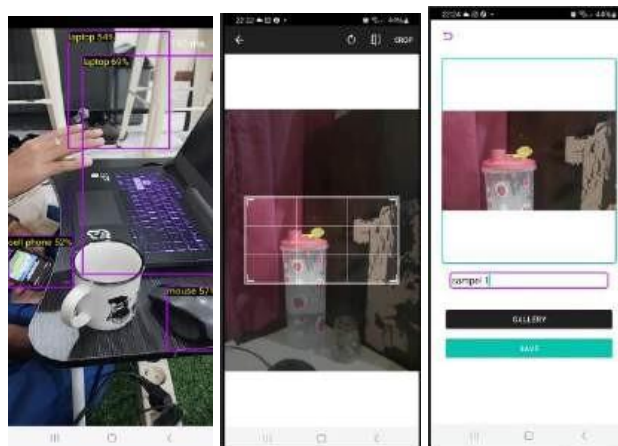
Activity bertindak sebagai launcher dan menyimpan session login dan tersimpan dalam session database firebase, selama user tidak melakukan logout akun. User tetap bisa menggunakan aplikasi tanpa harus melakukan login walaupun sudah melakukan close apps di histori.



Gambar 8 Activity Menu Utama

3.4.3 Activity Fitur Analis Object Detection

Activity Fitur Analisis Citra pada MainActivity terdapat 3 xml overlay cropImageLauncher untuk mengambil sampel objek secara langsung menggunakan kamera, xml history yaitu menampilkan hasil cropImageLauncher yang akan di save di dalam database dan akan menampilkan hasil analisis object berdasarkan class objek yang ada pada model.



Gambar 9 Activity Fitur Analis Object Detection

Overlay CropImageLauncher adalah sebuah fitur yang diaktifkan saat tombol galeri diklik, berfungsi untuk memilih citra yang akan dianalisis oleh model TensorFlow Lite. Proses analisis citra dilakukan ketika pengguna menekan tombol history, yang memulai pengolahan data pada model TFLite. Selain itu, fitur camera pada file XML memungkinkan deteksi objek secara langsung dengan menggunakan TFLite versi GPU 2.9.0. Proses ini menghasilkan bounding box yang mendeteksi objek secara realtime dan memberikan hasil yang akurat untuk kebutuhan analisis citra.



4. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan sistem deteksi wajah dan objek menggunakan algoritma Fast R- CNN yang mampu mengatasi berbagai tantangan dalam teknologi pengenalan wajah, seperti variasi pose, pencahayaan dan tumpang tindik objek. Deteksi wajah dan objek memainkan peran penting dalam berbagai aplikasi, seperti keamanan, autentikasi biometric dan sistem pengawasan. Namun, tantangan ini sering kali menyebabkan akurasi sistem menurun, terutama dalam aplikasi secara real time. Penelitian ini mengidentifikasi bahwa meskipun faster R-CNN telah digunakan secara luas untuk deteksi wajah dan objek dengan capaian hasil yang menunjukkan bahwa metode ini mampu mendeteksi objek dengan tingkat akurasi yang tinggi. Kecepatan pemrosesan yang dimiliki Fast R-CNN menjadikannya cocok untuk pengujian dataset secara real-time dalam aplikasi seperti keamanan, pengawasan, dan sistem otonom. Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan teknologi deteksi objek yang lebih andal dan efektif di berbagai bidang kehidupan.

UCAPAN TERIMAKASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Institut Teknologi PLN atas fasilitas, dukungan dan kesempatan yang diberikan selama proses penelitian ini berlangsung. Terima kasih kepada dosen pembimbing yang telah memberikan arahan dan bimbingannya sepanjang proses penelitian dan juga kepada teman-teman yang telah membantu dalam menyelesaikan penelitian ini. Kami berharap hasil dari penelitian ini dapat memberikan kontribusi bagi pengembangan ilmu pengetahuan dan teknologi di lingkungan Institut Teknologi PLN.

DAFTAR PUSTAKA

- [1] J. PARDEDE and H. HARDIANSAH, "Deteksi Objek Kereta Api menggunakan Metode Faster R-CNN dengan Arsitektur VGG 16," MIND J., vol. 7, no. 1, pp. 21–36, 2022, doi: 10.26760/mindjournal.v7i1.21-36.
- [2] R. World and D. Image, "What is Object ?," 2020, [Online]. Available: <https://www.mathworks.com/discovery/object-detection.htm>
- [3] C. Baretina, S. Saripudin, R. Listiana, and E. Damayanti, "Rancang Bangun Sistem Smart Door Lock Menggunakan Deteksi Wajah," J. Informatics Electron. Eng., vol. 01, no. 02, pp. 42–48, 2021.
- [4] D. T. Devi, V. A. Santos, and P. Rosyani, "Analisa Penggunaan Metode Faster R-CNN dalam Pengenalan Wajah: Systematic Literature Review," Bul. Ilm. Ilmu Komput. dan Multimed., vol. 1, no. 1, pp. 258–262, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/biikma>
- [5] Honainah, "Penerapan Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Deteksi Otomatis Interaksi Laki-Laki dan Perempuan," Nusantara J. Comput. Its Appl., vol. 7, no. 1, pp. 1–10, 2022.
- [6] M. Y. A. Thoriq, I. A. Siradjuddin, and K. E. Permana, "Deteksi Wajah Manusia Berbasis One Stage Detector Menggunakan Metode You Only Look Once (Yolo)," J. Teknoinfo, vol. 17, no. 1, p. 66, 2023, doi: 10.33365/jti.v17i1.1884.
- [7] J. I. Komputasi, V. No, M. Ssd, V. Mobilenet, and S. Model, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih," J. Ilm. Komputasi, vol. 19, no. 3, pp. 421–430, 2020, doi: 10.32409/jikstik.19.3.68.
- [8] Muhammad Fernanda Naufal Fathoni, Eva Yulia Puspaningrum, and Andreas Nugroho Sihananto, "Perbandingan Performa Labeling Lexicon InSet dan VADER pada Analisa Sentimen Rohingya di Aplikasi X dengan SVM," Modem J. Inform. dan Sains Teknol., vol. 2, no. 3, pp. 62–76, 2024, doi: 10.62951/modem.v2i3.112.
- [9] Suripto, R. N. Rahmanita, and A. S. Kirana, "Teknik pre-processing dan classification dalam data science," Binus University Graduate Program. [Online]. Available: [https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/#:~:text=Pengertian lain menyebutkan bahwa data,awal sebelum melakukan data mining.](https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/#:~:text=Pengertian%20lain%20menyebutkan%20bahwa%20data,awal%20sebelum%20melakukan%20data%20mining.)
- [10] S. Hanyfah, G. R. Fernandes, and I. Budiarto, "Penerapan Metode Kualitatif Deskriptif Untuk Aplikasi Pengolahan Data Pelanggan Pada Car Wash," Semnas Ristek (Seminar Nas. Ris. dan Inov. Teknol., vol. 6, no. 1, pp. 339–344, 2022, doi: 10.30998/semnasristek.v6i1.5697.
- [11] M. Tamba, "Analisis dan Implementasi Teknologi Deep Learning dalam Pengolahan Citra Digital," pp. 1–7.
- [12] S. T. E. Putri and A. Fahrurrozi, "Pendeteksian Objek Pada Citra Hewan Karnivora Dan Herbivora Menggunakan Faster R-Cnn," J. Ilm. Inform. Komput., vol. 27, no. 1, pp. 32–42, 2022, doi: 10.35760/ik.2022.v27i1.5858.