

BAB III

METODE PENELITIAN

Bab ini menjelaskan tahapan-tahapan sistematis yang digunakan dalam proses penelitian. Tahapan sistematis dimulai dari perancangan alat dan bahan, pengumpulan serta pengolahan data, hingga pelatihan dan evaluasi model. Setiap prosedur dijabarkan secara terstruktur untuk mendukung pengembangan model deteksi objek berbasis citra termal.

3.1 Perancangan Alat dan Bahan

Penelitian ini memerlukan dukungan perangkat keras, perangkat lunak, serta *dataset* citra termal yang digunakan dalam proses pelatihan dan pengujian model. Pada subbab ini dijelaskan secara rinci alat dan bahan yang digunakan selama proses penelitian berlangsung.

3.1.1 Perangkat Keras

Adapun perangkat keras yang digunakan untuk menunjang keberlangsungan penelitian ini yaitu:

1. GPU NVIDIA RTX 3060, VRAM 8GB
2. CPU Intel Core i7-10700F
3. 32GB RAM
4. Kamera termal

3.1.2 Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk menunjang keberlangsungan penelitian ini yaitu:

1. Windows 11
2. Visual Studio Code

Bahasa pemrograman yang digunakan pada penelitian ini adalah bahasa pemrograman Python versi 3.10

3.1.3 Lokus Penelitian

Penelitian ini mengimplementasikan pendekatan kuantitatif dengan desain eksperimen terstruktur untuk menilai kinerja model TRNet-128. Lokus penelitian ini difokuskan pada skenario kendaraan listrik otonom yang beroperasi di lingkungan terbatas, seperti kawasan industri tertutup, area perumahan, atau area kampus yang memiliki lalu lintas ringan dan mayoritas terdiri dari pejalan kaki serta pengendara sepeda motor. Lingkungan ini dipilih karena mencerminkan skenario nyata yang relevan dengan *dataset* yang digunakan, serta potensi aplikasi sistem deteksi objek termal secara akurat dalam kondisi minim cahaya.

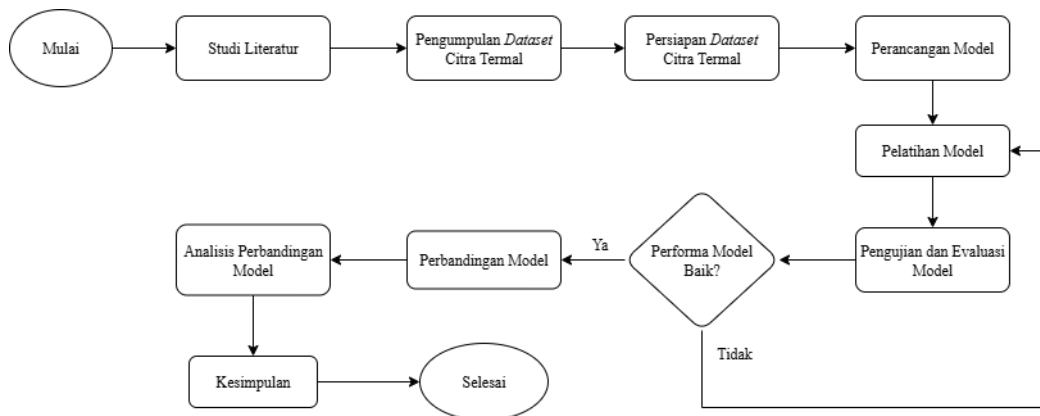
3.1.4 Dataset

Dataset yang digunakan merupakan kumpulan citra termal *grayscale* yang telah diberi anotasi. *Dataset* terdiri dari dua kelas objek yaitu manusia dan pengendara sepeda motor. *Dataset* ini dibagi menjadi tiga bagian, yaitu:

1. 1.008 gambar termal
2. Gambar berformat png dan anotasi berformat xml mengikuti standar Pascal VOC

3.2 Alur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sistematis yang direpresentasikan dalam diagram alur penelitian pada Gambar 3.1. Setiap tahapan memiliki peran penting dalam mendukung keberhasilan pengembangan dan evaluasi model TRNet-128.



Gambar 3.1 Diagram Alur Penelitian

Gambar 3.1 menunjukkan alur penelitian yang dimulai dari studi literatur, pengumpulan dan persiapan *dataset*, perancangan model, hingga tahap pelatihan, pengujian, dan analisis perbandingan model. Model TRNet-128 dibandingkan dengan model CNN sederhana guna mengevaluasi sejauh mana arsitektur kompleks dapat meningkatkan akurasi dan kemampuan generalisasi dibandingkan model dasar.

3.2.1 Studi Literatur

Penelitian ini dimulai dengan analisis terhadap studi sebelumnya yang relevan, mencakup metode, model arsitektur CNN, penggunaan *residual block*, serta pendekatan pemrosesan citra termal. Literatur ini berfungsi sebagai referensi dalam merumuskan kerangka teoritis.

3.2.2 Pengumpulan *Dataset* Citra Termal

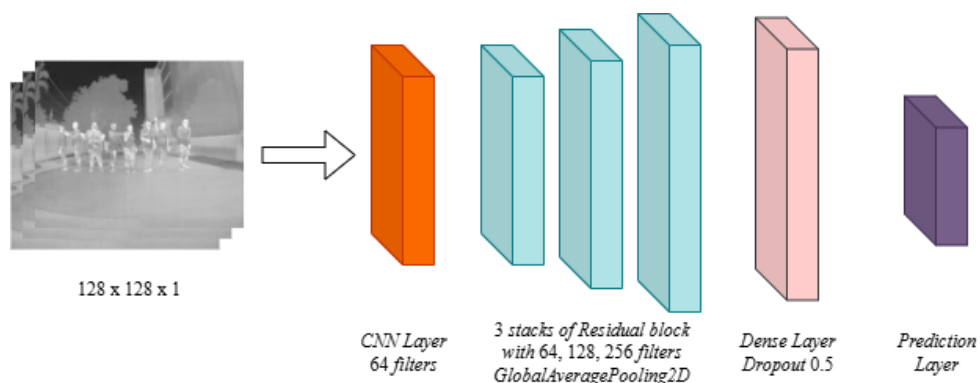
Dataset citra termal diperoleh melalui pengambilan gambar langsung di lingkungan Badan Riset dan Inovasi Nasional (BRIN) Gedung Samaun Samadikun menggunakan kamera termal. Pengambilan data dilakukan pada malam hari untuk menciptakan kondisi pencahayaan yang rendah sehingga mendekati skenario nyata penggunaan sistem pengenalan objek berbasis citra termal. *Dataset* ini selanjutnya digunakan untuk pelatihan dan pengujian model TRNet-128.

3.2.3 Persiapan *Dataset* Citra Termal

Dataset yang telah dikumpulkan akan memasuki proses seleksi gambar untuk memastikan setiap citra memiliki variasi yang cukup dan tidak terlalu mirip, agar model tidak mengalami *overfitting*. *Dataset* ini terdiri atas 1.008 gambar, dengan 50 gambar dipilih secara khusus sebagai data pengujian, sementara sisanya digunakan sebagai data pelatihan dan validasi. Data latih dan validasi berada di direktori yang sama dan akan dibagi secara otomatis menggunakan program Python dengan proporsi 80% untuk pelatihan dan 20% untuk validasi. Selanjutnya, seluruh gambar akan di-*resize* ke ukuran 128×128 *pixel* dan dilakukan normalisasi nilai *pixel* ke dalam rentang 0–1 untuk mempermudah proses pelatihan model. Proses anotasi dilakukan menggunakan format Pascal VOC (.xml), yang memuat informasi label serta koordinat *bounding box* dari objek yang terdapat dalam citra.

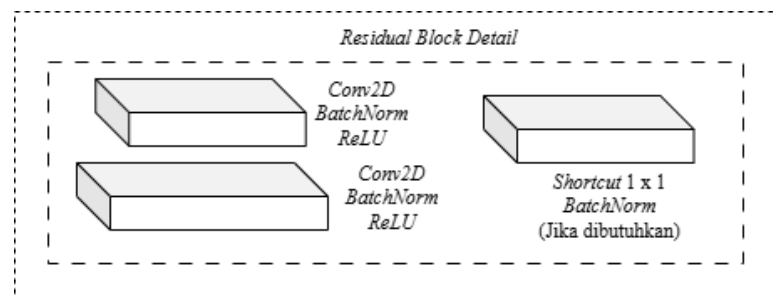
3.2.4 Perancangan Model

Perancangan model dilakukan dengan mengembangkan arsitektur TRNet-128 yang terdiri dari beberapa lapisan CNN yang dikombinasikan dengan *residual blocks* untuk mempertahankan kestabilan selama pelatihan dan menghindari masalah degradasi performa pada jaringan yang dalam. Model ini dirancang untuk bekerja secara optimal pada citra termal *grayscale* dengan resolusi *input* sebesar 128×128 *pixel*, serta untuk menyelesaikan dua tugas utama sekaligus, yaitu deteksi dan klasifikasi objek.



Gambar 3.2 Diagram Blok Model TRNet-128

Gambar 3.2 menggambarkan diagram blok arsitektur TRNet-128 secara umum. Dimulai dari citra *input* yang melewati beberapa lapisan CNN dan normalisasi. Kemudian dilanjutkan dengan serangkaian *residual blocks* yang bertujuan memperkuat fitur dan mencegah hilangnya informasi penting. Fitur yang dihasilkan kemudian digunakan baik untuk menentukan posisi objek dalam citra (deteksi) maupun mengklasifikasikan jenis objek. Setelah itu, fitur dikompresi melalui lapisan *Global Average Pooling* (GAP) dan dilanjutkan ke *fully connected layer* sebelum akhirnya menghasilkan prediksi kelas.



Gambar 3.3 Detail *Residual Block*

Gambar 3.3 memperlihatkan struktur detail dari *residual block* yang digunakan dalam model ini. *Residual block* terdiri dari dua lapisan konvolusi yang masing-masing diikuti dengan normalisasi *batch* dan aktivasi ReLU. Ciri khas utama dari blok ini adalah adanya *skip connection* yang menjumlahkan *input* asli dengan *output* dari dua lapisan konvolusi tersebut, lalu diteruskan melalui fungsi aktivasi. Teknik ini terbukti efektif dalam memperbaiki propagasi gradien dan mempertahankan informasi fitur dari lapisan sebelumnya.

Tabel 3.1 Arsitektur Model TRNet-128

<i>Layer</i>	<i>Output Shape</i>	<i>Kernel Size / Stride</i>	<i>Filters</i>	<i>Parameters</i>
<i>Input</i>	(128, 128, 1)	-	-	0
<i>Conv2D</i>	(64, 64, 64)	7×7 / 2	64	3.200
<i>BatchNormalization</i>	(64, 64, 64)	-	-	256

Layer	Output Shape	Kernel Size / Stride	Filters	Parameters
<i>ReLU</i>	(64, 64, 64)	-	-	0
<i>MaxPooling2D</i>	(32, 32, 64)	3×3 / 2	-	0
<i>Residual Block 1: Conv2D</i>	(32, 32, 64)	3×3 / 1	64	37.248
<i>BatchNormalization</i>	(32, 32, 64)	-	-	256
<i>ReLU</i>	(32, 32, 64)	-	-	0
<i>Conv2D</i>	(32, 32, 64)	3×3 / 1	64	36.928
<i>BatchNormalization</i>	(32, 32, 64)	-	-	256
<i>Add</i>	(32, 32, 64)	-	-	0
<i>ReLU</i>	(32, 32, 64)	-	-	0
<i>Residual Block 2: Conv2D</i>	(16, 16, 128)	3×3 / 2	128	73.856
<i>BatchNormalization</i>	(16, 16, 128)	-	-	512
<i>ReLU</i>	(16, 16, 128)	-	-	0
<i>Conv2D</i>	(16, 16, 128)	3×3 / 1	128	147.584
<i>BatchNormalization</i>	(16, 16, 128)	-	-	512
<i>Conv2D (Shortcut)</i>	(16, 16, 128)	1×1 / 2	128	8.448
<i>BatchNormalization</i>	(16, 16, 128)	-	-	512
<i>Add</i>	(16, 16, 128)	-	-	0
<i>ReLU</i>	(16, 16, 128)	-	-	0
<i>Residual Block 3: Conv2D</i>	(8, 8, 256)	3×3 / 2	256	295.168
<i>BatchNormalization</i>	(8, 8, 256)	-	-	1.024

Layer	Output Shape	Kernel Size / Stride	Filters	Parameters
<i>ReLU</i>	(8, 8, 256)	-	-	0
<i>Conv2D</i>	(8, 8, 256)	3×3 / 1	256	590.080
<i>BatchNormalization</i>	(8, 8, 256)	-	-	1.024
<i>Conv2D (Shortcut)</i>	(8, 8, 256)	1×1 / 2	256	33.280
<i>BatchNormalization</i>	(8, 8, 256)	-	-	1.024
<i>Add</i>	(8, 8, 256)	-	-	0
<i>ReLU</i>	(8, 8, 256)	-	-	0
<i>GlobalAveragePooling2D</i>	(256)	-	-	0
<i>Dense</i>	(512)	-	-	131.584
<i>Dropout</i>	(512)	-	-	0
<i>Dense</i>	(# Classes)	-	-	(# Classes × 513)

Tabel 3.1 berisi struktur lengkap dari arsitektur model TRNet-128. Model ini diawali dengan lapisan Conv2D berukuran *kernel* 7×7 dan *stride* 2 yang digunakan untuk menangkap fitur awal dari citra termal beresolusi 128×128 *pixel*. Ukuran *kernel* yang besar di awal bertujuan untuk menjangkau konteks spasial yang luas secara cepat. Lapisan ini diikuti oleh *batch normalization* dan fungsi aktivasi ReLU untuk mempercepat konvergensi dan menghindari *vanishing gradient*. Proses ini kemudian diikuti oleh *max pooling* untuk mengurangi dimensi spasial dan kompleksitas komputasi.

Model kemudian dilanjutkan dengan tiga *residual blocks* yang terdiri dari dua lapisan konvolusi masing-masing. Pada *residual block* pertama, dimensi *output* tetap, sehingga *shortcut connection* langsung dapat dijumlahkan tanpa transformasi. Pada *residual block* kedua dan ketiga, terjadi perubahan dimensi yaitu peningkatan jumlah filter dan *downsampling*, sehingga *shortcut* dilengkapi dengan konvolusi 1×1 untuk mencocokkan dimensi. *Residual blocks* ini memberikan

kemampuan jaringan untuk mempelajari representasi yang lebih dalam tanpa kehilangan informasi fitur penting.

Setelah *residual block* terakhir, dilakukan GAP untuk merangkum fitur secara keseluruhan dari hasil konvolusi. Hasil pooling dikirim ke lapisan *dense* sebanyak 512 unit yang dilengkapi dengan *dropout* untuk mengurangi *overfitting*. Selanjutnya *output* dikirim ke lapisan *dense* akhir dengan jumlah unit sesuai dengan jumlah kelas, dan fungsi aktivasi *softmax* digunakan untuk menghasilkan probabilitas klasifikasi.

3.2.5 Pelatihan Model

Proses pelatihan model dilakukan dengan menggunakan data pelatihan dan validasi yang telah diproses dalam format TensorFlow *Dataset*. Data dikemas dalam bentuk *batch* dengan ukuran 32 yang merupakan jumlah citra yang diproses secara bersamaan dalam satu iterasi untuk mempercepat pelatihan dan efisiensi penggunaan memori.

Pelatihan dilakukan selama 10 *epoch*, yaitu jumlah siklus lengkap di mana seluruh *dataset* pelatihan digunakan sekali untuk memperbarui bobot model. Model dilatih menggunakan *optimizer* Adam karena kemampuannya yang adaptif dalam menyesuaikan *learning rate*, serta *loss function* berupa *sparse categorical crossentropy* yang sesuai untuk klasifikasi multi-kelas dengan label numerik.

Proses pelatihan mencakup pelacakan metrik akurasi dan *loss* pada data pelatihan dan validasi untuk mengetahui performa dan menghindari *overfitting*. Hasil pelatihan disimpan dalam bentuk file model dengan ekstensi h5 serta disertai grafik riwayat pelatihan yang berisi akurasi dan *loss* terhadap *epoch* yang otomatis disimpan dalam folder *output*. Riwayat pelatihan ini menjadi dasar untuk mengevaluasi kestabilan proses pembelajaran model.

Selain itu, pengelompokan label dan *encoding* label dilakukan secara otomatis menggunakan *dictionary mapping* dari label ke indeks numerik. Seluruh pelatihan dan evaluasi berjalan secara otomatis pada *pipeline* program Python yang telah disusun, sehingga hasil pelatihan dapat direproduksi dan dianalisis secara konsisten.

Miftahul Falah, 2025

PENGEMBANGAN MODEL THERMAL RESIDUALNET-128 UNTUK DETEKSI OBJEK DALAM CITRA THERMAL MENGGUNAKAN PEMBELAJARAN RESIDUAL

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3.2.6 Pengujian dan Evaluasi Model

Melakukan pengujian model terhadap data uji untuk mengevaluasi kemampuan generalisasi model. Proses pengujian dilakukan dengan memanfaatkan 50 citra termal yang telah dipisahkan khusus untuk pengujian. Model yang telah dilatih digunakan untuk memprediksi label dan posisi objek dari setiap citra tersebut. Pengujian dilakukan secara otomatis melalui program Python yang digunakan untuk membaca gambar uji beserta file anotasinya serta memvisualisasikan hasil prediksi *bounding box* serta label pada setiap gambar. Proses ini memanfaatkan beberapa fungsi untuk memuat, memproses, dan menampilkan hasil deteksi. *Output* dari proses pengujian disimpan dalam direktori hasil dalam bentuk gambar dengan anotasi visual.

Selain visualisasi, model juga diuji dengan menghasilkan prediksi terhadap data validasi untuk menghitung metrik evaluasi seperti *confusion matrix* dan *classification report* yang berisi *accuracy*, *precision*, *recall*, dan *F1-score*. Data hasil prediksi digunakan pula untuk menghitung *mean Average Precision* (mAP) yang merupakan metrik penting dalam evaluasi model deteksi objek. Semua hasil pengujian, termasuk grafik, *matrix*, dan skor evaluasi disimpan dalam file Excel dan gambar PNG secara otomatis oleh sistem. Jika performa belum baik, dilakukan evaluasi ulang terhadap hasil pengujian dan parameter pelatihan yang digunakan. Langkah ini dapat mencakup penyesuaian arsitektur model, parameter *hyperparameter*, atau pengimplementasian teknik augmentasi pada *dataset*. Jika hasil evaluasi menunjukkan bahwa model telah mencapai kriteria performa yang diharapkan, maka proses dilanjutkan ke tahap perbandingan model untuk melihat keunggulan relatif dari TRNet-128 dibandingkan pendekatan lainnya.

3.2.7 Perbandingan Model

Membandingkan performa TRNet-128 dengan model lain atau *baseline* untuk mengetahui keunggulan yang dimiliki. Dalam penelitian ini, TRNet-128 akan dibandingkan secara langsung dengan model CNN sederhana yang tidak menggunakan *residual block*. Model pembanding dirancang dengan struktur konvolusi dasar tanpa adanya jalur *shortcut*, sehingga memungkinkan peneliti

Miftahul Falah, 2025

PENGEMBANGAN MODEL THERMAL RESIDUALNET-128 UNTUK DETEKSI OBJEK DALAM CITRA THERMAL MENGGUNAKAN PEMBELAJARAN RESIDUAL

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

untuk mengamati dampak penggunaan *residual block* terhadap akurasi klasifikasi dan efektivitas deteksi objek.

Perbandingan dilakukan menggunakan metrik evaluasi yang sama, seperti *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, dan *mean Average Precision* (mAP), baik pada data validasi maupun data uji. Hal ini bertujuan untuk memberikan gambaran menyeluruh mengenai kelebihan arsitektur TRNet-128 dalam mengenali dan mendeteksi objek pada citra termal dibandingkan arsitektur konvensional yang lebih dangkal.

3.2.8 Analisis Perbandingan Model

Menganalisis hasil perbandingan secara menyeluruh untuk menilai keefektifan model dalam tugas klasifikasi dan deteksi objek termal. Analisis ini mencakup interpretasi nilai metrik yang diperoleh dari masing-masing model, seperti *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, dan *mean Average Precision* (mAP), untuk mengidentifikasi kekuatan dan kelemahan dari masing-masing pendekatan. Selain itu, dilakukan pula pengamatan terhadap hasil visualisasi prediksi objek yang dihasilkan oleh masing-masing model guna mengetahui kualitas deteksi secara kasat mata pada citra termal. Dengan demikian, analisis ini memberikan pemahaman mendalam terhadap performa relatif TRNet-128 dibandingkan CNN sederhana dari berbagai aspek.