



# Segmentasi Objek Berbasis Gambar Termal Menggunakan *Deep Learning (Pre-Trained Resnext 50)*

## *Object Segmentation Based On Thermal Images Using Deep Learning (Pre-Trained Resnext 50)*

R. Aldam Dwi Fauzan<sup>1</sup>, Arief Suryadi Satyawan<sup>2</sup>, Sri Desy Siswanti<sup>3</sup>, Heni Puspita<sup>4</sup>

<sup>1</sup>Teknik Elektro, Universitas Nurtanio Bandung

*E-mail: aldamdweifauzan55@gmail.com*

<sup>2</sup>Badan Riset dan Inovasi Nasional

*E-mail: arief.suryadi@akane.waseda.jp*

<sup>3</sup>Universitas Nurtanio Bandung

*E-mail: desy0712unnur@gmail.com*

<sup>4</sup>Universitas Nurtanio Bandung

*E-mail: Henipuspita043@gmail.com*

**Abstract—** The transportation sector at this time has experienced many technological developments which have been well received by the public, especially the people of Indonesia. Along with the development of transportation technology has undergone many developments, with sophistication and increased comfort and better security. So Autonomous Car technology was created that can help drivers to maintain safety while driving. Autonomous car was built using the Neural Network control method, and also Image Processing as signal processing with image input, and with a flip camera used for vehicle input data. Autonomous cars have many positive impacts on human life today, so humans can minimize time properly. Travel safety is maintained, and can be more productive when driving. The method that is currently developing rapidly is automatic extraction using deep learning. In this final project, automatic extraction method with deep learning technology used is Fully Convolutional Network (FCN) with Residual Neural Network Next (ResNext) architecture. In this study, the extraction accuracy for automatic vehicle function training reached 98% for ResNext 50 with a resolution of 640x540 pixels. Semantic segmentation will then test with 34030 image frames offline. In ResNext 50 architecture contains 20512 frames in good category, 7883 in adequate category and 5605 in poor category.

**Keywords:** Deep Learning, Fully Convolutional Network, Image Processing, Neural Network

**Abstrak—** Bidang transportasi pada masa kini telah mengalami banyak perkembangan teknologi yang disambut dengan baik oleh masyarakat khususnya masyarakat Indonesia. Dengan seiring perkembangan teknologi transportasi sudah mengalami banyak perkembangan, dengan kecanggihan serta peningkatan kenyamanan dan keamanan yang lebih baik. Maka dibuatlah teknologi *Autonomus Car* yang mampu membantu para pengendara untuk menjaga keselamatan saat berkendara. *Autonomus car* dibangun dengan menggunakan metode kontrol *Neural Network*, dan juga *Image Processing* sebagai pemrosesan sinyal dengan input berupa gambar, dan dengan kamera flir digunakan untuk data input kendaraan. *Autonomus car* memiliki banyak dampak positif di kehidupan manusia hari ini, maka manusia dapat meminimalisir waktu dengan baik. Keselamatan perjalanan terjaga, serta mampu lebih produktif saat berkendara. Metode yang saat ini berkembang dengan pesat adalah ekstraksi otomatis menggunakan *deep learning*. Dalam Tugas

Akhir kali ini menggunakan metode ekstraksi otomatis dengan teknologi *deep learning* yang digunakan adalah *Fully Convolutional Network* (FCN) dengan arsitektur *Residual Neural Network Next* (ResNext). Penelitian kali ini, akurasi ekstraksi untuk pelatihan fungsi kendaraan otomatis mencapai 98% untuk ResNext 50 dengan resolusi yang dipakai 640x540 pixel. Segmentasi semantic selanjutnya akan menguji dengan 34030 frame gambar secara offline. Pada arsitektur ResNext 50 terdapat 20512 frame kategori baik, 7883 kategori cukup dan 5605 kategori kurang baik.

Kata Kunci : Deep Learning, Fully Convolutional Network, Image Processing, Neural Network

## I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi kini senantiasa mengalami peningkatan pesat pada pemanfaatannya yang semakin beragam menyesuaikan dengan kegiatan-kegiatan manusia yang modern dan canggih, khususnya di Indonesia sehingga memungkinkan untuk dapat diaplikasikan ke berbagai bidang, termasuk mendorong perkembangan era industri 4.0. khususnya pada bidang ilmu pengetahuan dan sistem komunikasi. Pada saat ini diberbagai negara sudah banyak yang mulai mengembangkan berbagai teknologi, salah satunya adalah kendaraan listrik yang bisa tanpa menggunakan tenaga manusia atau yang bisa dikenal dengan kendaraan listrik otonom (KLO). Kendaraan yang satu ini bisa mengurangi kelalaian manusia dalam berkendara dan juga kendaraan ini bisa memudahkan manusia dalam bepergian kemanapun tanpa perlu mengendarai kendaraannya. Oleh karena itu kendaraan listrik otonom memerlukan sistem yang dapat mendeteksi serta membedakan objek dengan menggunakan *sensor camera thermal* dengan metode segmentasi berbasis *Deep Learning* yang dilatih dengan menggunakan dataset yang dibutuhkan.

Dengan ini, penulis menguji sistem pendeteksi berbasis *sensor camera thermal* menggunakan kamera FLIR ADK yang akan dikembangkan sesuai dengan klasifikasinya sehingga dapat mengsegmentasikan gambar. Perangkat *camera thermal* yang digunakan pada saat ini akan dibantu dengan *mini computer Artificial Intelligence* (AI) untuk mesin otonom yang bernama Jetson AGX Xavier sebagai tempat pengolahan data.

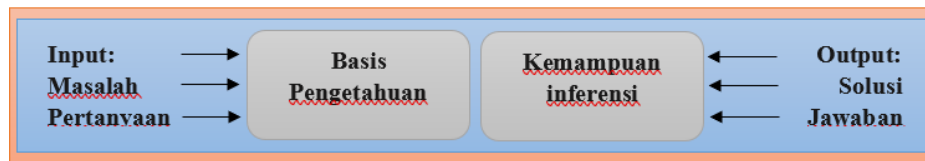
## II. LANDASAN TEORI

### A. Artificial Intelligence

Kecerdasan buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Maka dari itu AI dapat membantu manusia untuk menyelesaikan masalah, karena AI mencakup bidang yang cukup besar. Contoh dari kecerdasan buatan itu sendiri diantaranya yaitu *Virtual Reality* (VR), Aplikasi ojek online, drone, e-commerce, platform google, mobil pintar, serta beberapa negara sudah mengembangkan rumah pintar. Untuk mewujudkan aplikasi AI diperlukan dua bagian utama, yaitu:

- a. Basis pengetahuan (*knowledge base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Kemampuan inferensi (*inference*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

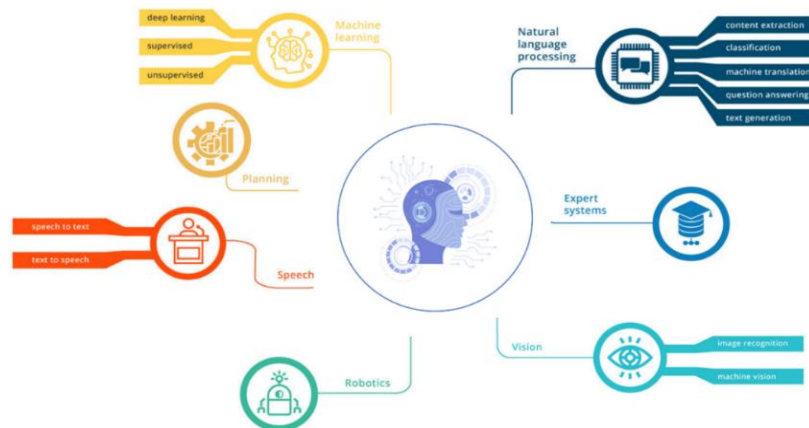
Penerapan konsep kecerdasan buatan pada komputer dapat dilihat pada Gambar 1.



**Gambar 1** Penerapan Konsep Kecerdasan Buatan

Dari Gambar 1 menjelaskan bahwa basic awal dari AI yaitu, masalah dari pertanyaan adalah sebagai input dan basis pengetahuan sebagai proses. Kemudian kemampuan inferensi sebagai proses yang akan menuju output seperti solusi dan jawaban.

Dari Gambar 2, AI biasanya mencakup banyak elemen, seperti *machine learning*, *natural language processing (NLP)*, *expert systems*, *speech*, *vision*, *planning*, dan *robotics*. Dalam studi ini, penulis melihat penelitian *machine learning* dan fokus pada pembelajaran *deep learning*, tetapi AI juga dapat mencakup pendekatan yang tidak memerlukan pembelajaran.



**Gambar 2** Himpunan Bagian Artificial Intelligence

## B. Machine Learning

*Machine learning* merupakan salah satu cabang dari *Artificial Intelligence* (AI) atau kecerdasan buatan. Ini merupakan sistem yang bisa mengadaptasi kemampuan manusia untuk belajar yang tanpa kita sadari penggunaannya sudah sering hadir di kehidupan sehari-hari. Ilmu ini berfokus untuk membuat sistem atau algoritma yang terus belajar dari data dan meningkatkan akurasi dari waktu ke waktu tanpa pemrograman tertentu. Dalam aplikasi *machine learning*, algoritma atau urutan proses statistik dilatih untuk menemukan pola dan fitur tertentu dalam jumlah data yang besar. Hal ini bertujuan untuk membuat suatu keputusan maupun prediksi berdasarkan data-data tersebut.

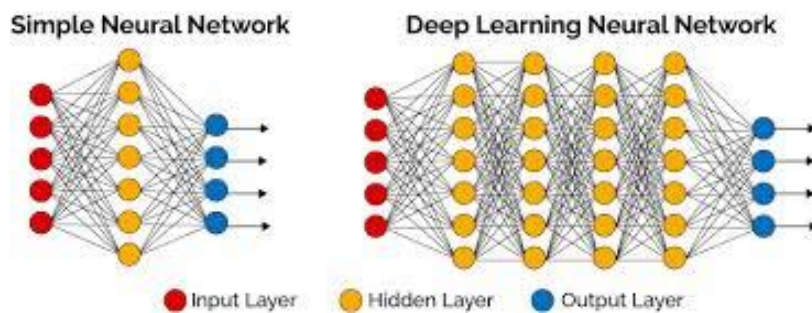
*Machine learning* bermanfaat untuk menyelesaikan permasalahan dunia dengan cara yang terukur. Aplikasi ilmu kecerdasan buatan ini pun bisa digunakan pada berbagai macam industri dan terus digunakan oleh pemilik industri besar dan peneliti agar dapat terus berkembang. Terdapat beberapa permasalahan yang dapat diselesaikan dengan menggunakan *machine learning*. Berikut merupakan beberapa permasalahan yang pada umumnya diselesaikan oleh *machine learning*.

1. *Classification*
2. Deteksi Anomali
3. Estimasi Fungsi Massa Probabilitas; dan lain sebagainya.
4. Mesin Translasi
5. *Regression*
6. Transkripsi

C. *Deep Learning*

*Deep Learning* adalah cabang dari *machine learning* yang algoritmanya terinspirasi dari struktur otak manusia. Struktur ini disebut jaringan saraf tiruan (JST). Ini pada dasarnya adalah jaringan saraf dengan tiga atau lebih lapisan JST. Itu dapat mempelajari dan mengadaptasi data dalam jumlah besar untuk memecahkan masalah yang sulit dipecahkan dengan algoritme pembelajaran mesin lainnya. *Deep Learning* terdiri dari beberapa jaringan saraf yang saling berhubungan. *Deep Learning* terdiri dari beberapa jaringan saraf yang saling berhubungan. Berikut beberapa algoritmanya:

1. *Convolutional Neural Network* (CNN)
2. *Recurrent Neural Network* (RNN)
3. *Long Short Term Memory Network* (LSTM)
4. *Self Organizing Maps* (SOM)



Gambar 3 Perbandingan Simple dan *Deep Learning Neural Network* [11].

D. *Image Processing*

*Image processing* adalah suatu bentuk pengolahan yang menggunakan teknik tertentu untuk mengambil input berupa pemrosesan sinyal atau suatu gambar dan mengubahnya menjadi gambar lain sebagai keluaran. Pengolahan gambar dilakukan untuk memberikan informasi kepada komputer dan manusia sebagai data baru yang dapat diolah menjadi fungsi dan model modern. Berikut adalah termasuk beberapa komponen pemrosesan gambar:

1. *Convolution* (konvolusi)

*Convolution* didefinisikan sebagai operasi dua fungsi. Dalam menganalisis citra, fungsi pertama merupakan nilai masukan atau nilai piksel di suatu posisi dalam citra dan fungsi kedua merupakan menjadikan sebuah matriks *filter* atau *kernel*. Nilai keluaran diperoleh dari perkalian titik dua fungsi. *Filter* kemudian digeser ke posisi berikutnya pada citra, yang kemudian disebut sebagai panjang *stride* [9].

$$Feature\ map = [W - F] + 1 \dots$$

Keterangan :

W = Ukuran Volume Gambar

F = Ukuran Filter

2. *Stride*

*Stride* adalah parameter yang menentukan berapa jumlah pergeseran *filter*. Jika nilai *stride* adalah 1, maka *filter* akan bergeser sebanyak 1 *pixels* secara *horizontal* lalu *vertical*. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar [9].

3. *Padding*

*Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan disetiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolution layer* (*Feature map*).

#### 4. *Pooling*

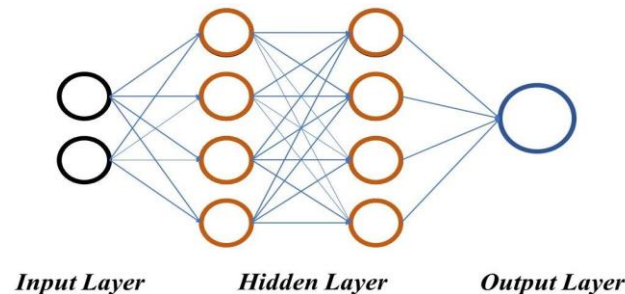
*Pooling layer* biasanya berada setelah *convolution layer*. Pada prinsipnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*.

#### 5. *Flatten*

*Output* dari proses konvolusi yaitu setelah dilakukan operasi *average pooling* masih berbentuk multidimensional *array* sehingga belum bisa dijadikan *input* pada *neural network*. *Inputan* pada neural network harus berbentuk vector satu dimensi.

#### 6. Fully Connected Layer

Lapisan terakhir dalam CNN adalah *fully connected layer* ilustrasinya dapat dilihat pada Gambar 4, yang berarti setiap *neuron* pada lapisan sebelumnya dihubungkan dengan setiap *neuron* di *fully connected layer*, seperti konvolusi, ReLU, dan *pooling layer*, memungkinkan terdapat satu atau lebih *fully connected layer* pada suatu arsitektur CNN.



Gambar 4 Fully Connected Layer

#### 7. *Activation Layer*

Pada *activation layer* terdapat beberapa fungsi aktivasi yang digunakan untuk menentukan apakah *neuron* dari jaringan tersebut akan diaktifkan atau dinonaktifkan. Secara umum terdapat 2 jenis fungsi aktivasi, *Linear* dan *Non-Linear Activation*. Salah satu fungsi aktivasi yaitu *Rectified linear unit* (ReLU) bersifat *non-linear* dengan kombinasi *linear* sehingga *layers* yang berbeda dapat ditumpuk bersama.

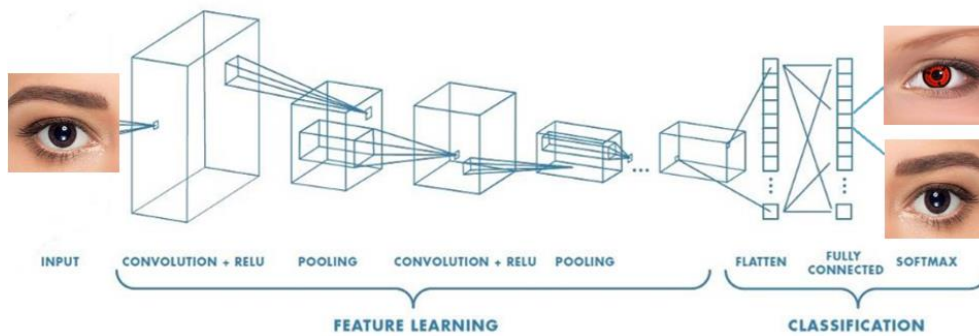
#### 8. *Softmax*

Fungsi *softmax* digunakan untuk metode klasifikasi dengan jumlah kelas yang banyak, seperti regresi logistik multinomial, ANN, dan lain sebagainya. *Softmax* merupakan sebuah fungsi yang mengubah nilai bobot keluaran dari semua proses di *fully connected layer* menjadi rentang nilai 0 dan 1.

### E. *Convolutional Neural Network (CNN)*.

CNN atau bisa disebut juga (Jaringan Saraf Konvolusi) adalah *multilayer feed-forward neural network* yang menggunakan persepsi untuk metode *supervised learning* dan analisa data. Algoritma ini terutama digunakan dengan data visual, seperti klasifikasi gambar. CNN terdiri dari banyak *layer* untuk memproses dan mengenali suatu objek. Ia biasanya digunakan untuk memproses gambar dan mendeteksi objek.

Dalam perkembangannya, ada banyak arsitektur CNN yang umum digunakan, antara lain: r-cnn, faster r-cnn, SSD (*single shot multibox detector*), Alexnet, Vggnet, Googlenet, Resnet, Resnext, dll.



Gambar 5 Struktur Dasar Convolutional Neural Network.

F. Fully Convolutional Network (FCN)

Fully Convolutional Network (FCN) adalah CNN tanpa koneksi lengkap (fully connected) dan merupakan ekstraksi fitur dengan segmentasi semantik. Proses ini menggunakan pendekatan FCN. Pendekatan ini bekerja dengan mengadaptasi dan memodifikasi model deteksi CNN agar sesuai untuk digunakan dalam proses segmentasi

G. NVIDIA Jetson AGX Xavier

NVIDIA AGX Jetson Xavier Developer Kit adalah kit pengembangan AI yang dapat digunakan untuk menjalankan berbagai kebutuhan AI modern dengan kinerja luar biasa. Dengan NVIDIA Jetson AGX Xavier, pengembang dapat menjalankan framework dan model AI untuk aplikasi pengenalan gambar, pengenalan objek, segmentasi, pengenalan suara, dan banyak lagi.

NVIDIA Jetson yang digunakan dalam tugas akhir ini adalah jenis NVIDIA Jetson AGXXavier. Spesifikasi berikut disertakan dalam perangkat NVIDIA Jetson AGXXavier pada Tabel 2.4

Table 1 Spesifikasi NVIDIA Jetson AGX Xavier

|                     |  |
|---------------------|--|
| <b>GPU</b>          | 512-core Volta GPU with Tensor Cores   |
| <b>CPU</b>          | 8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3  |
| <b>Memory</b>       | 32GB 256-Bit LPDDR4x   137GB/s   |
| <b>Storage</b>      | 32GB eMMC 5.1  |
| <b>Connectivity</b> | Gigabit Ethernet, M.2 Key E  |
| <b>Display</b>      | HDMI 2.0 and eDP 1.4   |
| <b>USB</b>          | 2x USB 3.1, DP (Optional), PD (Optional) Close-System Debug and Flashing SupPort on 1 Port |

H. Python

Python adalah bahasa pemrograman tingkat tinggi yang universal, interaktif, dan berorientasi objek. Itu dibuat oleh Guido van Rossum antara 1985 dan 1990. Filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Pendekatan pembangunan bahasa dan berorientasi objek dimaksudkan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala besar dan kecil. Bahasa pemrograman Python ini dapat digunakan di berbagai platform dengan filosofi desain yang berfokus pada tingkat keterbacaan kode dan merupakan salah satu bahasa paling populer yang terkait dengan Data Science, Machine Learning, dan Internet of Things (IoT) [14].

I. FLIR (Vision Automotive Development Kit) ADK Dokumen

FLIR (Vision Automotive Development Kit) ADK dokumen menjelaskan instruksi rinci tentang kondisi dan persyaratan untuk mengintegrasikan kamera FLIR ADK dengan berbagai metode komunikasi termasuk kedalam USB, ethernet papan aksesori gmsl untuk komunikasi, streaming video, dan sinkronisasi perangkat keras. Kamera FLIR ADK menyediakan long waver inferred image (lwir) yaitu 640x512 piksel. ini terasa radiasi termal dengan panjang gelombang 8-14um. Output-nya adalah 8bit atau 16bit gambar monokrom. Kamera FLIR ADK hadir dalam



tiga antarmuka berbeda: dengan keluaran USB, keluaran gmsl, dan keluaran gmsl dengan konverter untuk gmsl ke ethernet. FLIR ADK juga memiliki empat opsi lensa yang berbeda, pada tabel 2.5 diperlihatkan empat opsi lensa pada kamera FLIR ADK. Pada gambar 2.22 diperlihatkan kamera FLIR ADK[16].

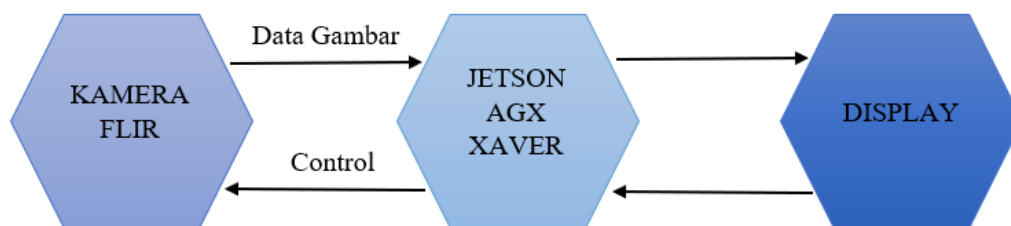
**Table 2** Table Kamera FLIR ADK Empat Opsi Lensa

| Horizontal FOV | Vertical FOV |
|----------------|--------------|
| 24             | 19.2         |
| 32             | 25.6         |
| 50             | 40           |
| 75             | 60           |

### III. METODE/MODEL YANG DIUSULKAN

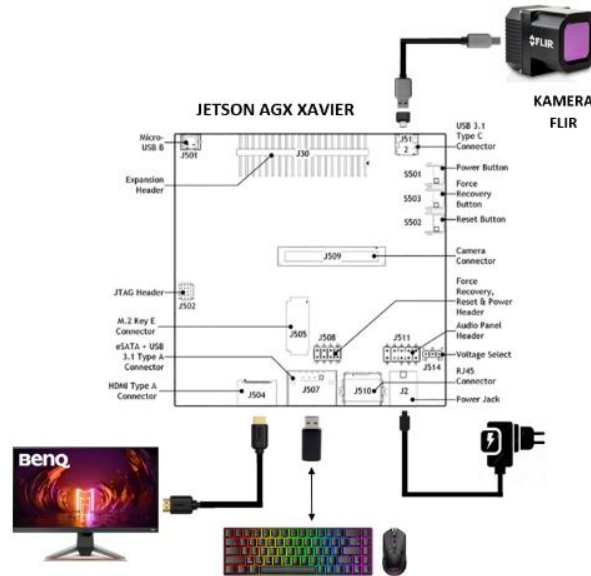
#### A. Desain

Desain sistem pendeteksian objek di lingkungan terbatas berbasis gambar *Thermal* menggunakan Arsitektur CNN Pre-trained *Resnext-50* terdiri dari dua bagian utama yaitu desain konstruksi *hardware* dan juga *software*. Perancangan pendeteksian objek menggunakan gambar *thermal* terdapat beberapa bagian yaitu desain konstruksi perangkat keras sampai dengan konstruksi perangkat lunak. Pada gambar 6 dapat dilihat ilustrasi perancangan *hardware* yang digunakan untuk pendeteksian objek secara segmentasi. Konstruksi ini menggunakan kamera *thermal* FLIR, Jetson AGX Xavier, Monitor.



**Gambar 6** Konstruksi *Hardware* Pendeteksian Objek berbasis Arsitektur CNN Pre-trained *Resnext-50*

Pada Gambar 7 merupakan gambar desain perangkat keras yang akan digunakan untuk pendeteksian objek berbasis gambar *thermal* menggunakan *deep learning pre-trained Resnext-50*. Pengolahan data yang dilakukan di terminal menggunakan python sebagai Bahasa yang mudah untuk dipahami, kemudian data yang telah diterima jetson melalui kamera akan diolah sehingga sehingga pendeteksian objek sesuai klasifikasinya dapat direalisasikan dengan baik.



Gambar 7 Ilustrasi Perancangan Hardware

B. Spesifikasi Hardware dan versi software

Table 3 Spesifikasi Hardware

| No | Hardware          | Spesifikasi  |
|----|-------------------|--|
| 1  | Kamera Flir ADK™  | Resolution : 640 x 512<br>Pixel Size : 12 μm<br>Frame Rate : 30 & 60 Hz Selectable<br>Thermal Imaging : Boson™ - Uncooled VOx<br>Power Consumption : 4W – 12W<br>Data Format : RAW 8bit – 16bit<br>Protection : IP67<br>Operating Temperature : -40 <sup>0</sup> F – 167 <sup>0</sup> F<br>Solar protection : Sun will not damage sensor |
| 2  | Jetson AGX Xavier | GPU : 512-core Volta GPU with Tensor Cores<br>CPU : 8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3<br>Memory : 32GB 256-Bit LPDDR4x   137GB/s<br>Storage : 32GB eMMC 5.1<br>USB : 2x USB 3.1, DP (Optional), PD (Optional) Close-System Debug and Flashing Support on 1 Port  |
| 3  | Komputer          | CPU : Intel Core I9-9900K 3.60GHz<br>GPU : Nvidia GeForce RTX 3080 10GB<br>Motherboard : Msi Z390-A PRO<br>RAM : 32GB Memory 3200Mhz<br>HDD : 1TB<br>SSD : 512 SSD NVME M2<br>Power Supply : 750W 80+ Gold Fully Modular   |
| 4  | Monitor           | Full HD 24 inch 75 Hz  |

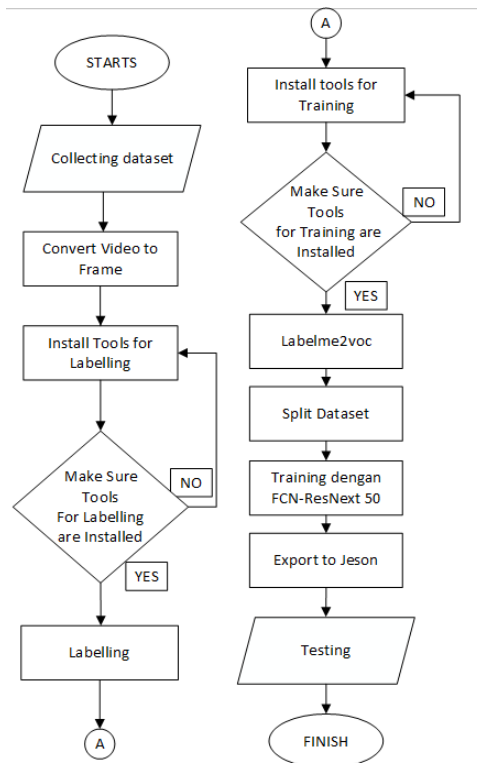
C. Langkah Alur Sistem Segmentasi Objek Berbasis Gambar Termal

Pada Gambar 8 dilihat langkah langkah alur untuk segmentasi objek berbasis gambar termal. Pengambilan gambar yang digunakan untuk dataset dilakukan di lingkungan Universitas Nurtanio Bandung menggunakan mobil sebagai pengaplikasian dan kamera Flir merupakan alat yang digunakan untuk mengambil data berupa video. Setelah video disimpan lalu diubah menjadi gambar untuk software yang digunakan yaitu software labeling ataupun training. Lalu gambar di



annotasi kan menggunakan *labeling* dengan data klasifikasi objek yang dibuat serta diaplikasikan pada *tools labelme* dengan format JSON.

Jika pemasangan *tools* untuk *training* telah selesai kemudian selanjutnya melakukan *training* pada gambar atau *dataset* yang telah di labeling, sebelumnya format gambar diubah dari JSON ke PNG dengan menggunakan program *labelme2voc* serta split gambar dataset tersebut menjadi *validation* dan *train* menggunakan bahasa pemrograman python sehingga proses *training* dengan arsitektur CNN pre-trained *Resnext-50* dapat dilakukan. File hasil training dimasukkan ke Jetson AGX Xavier untuk malakukan *testing*. Tahap terakhir adalah tahap pengujian yang dilakukan secara *real time* dan juga *offline* menggunakan model arsitektur CNN *Pre-trained Resnext-50*.



**Gambar 8** Flowchart Sistem Segementasi Objek

#### IV. HASIL/IMPLEMENTASI MODEL DAN PEMBAHASAN

Setelah proses pembuatan sistem segmentasi objek berbasis gambar thermal telah selesai, maka tahap selanjutnya yang harus dilakukan adalah uji fungsi untuk mengetahui apakah sistem deteksi tersebut dapat digunakan dengan baik, cukup atau kurang baik untuk mengetahui batasan dari sistem segmentasi tersebut, Terdapat dua pengujian dari software aplikasi segmentasi objek berbasis gambar thermal yang dilakukan.

Tabel 4 Tahapan Pengujian

| No | Tahap Pengujian   | Tujuan  |
|----|-------------------|---|
| 1  | Pengujian Pertama | Pengujian saat malam hari dengan <i>dataset</i> yang di labelin terlebih dahulu |
| 2  | Pengujian Kedua   | Pengujian saat siang hari dengan <i>dataset</i> yang di labelin terlebih dahulu |

##### A. Pengukuran Nilai Model

Pada bagian ini akan melakukan pengukuran hasil nilai model menggunakan *dataset* yang telah dilatih sebelumnya. Pada proses pengukuran ini harus melakukan *training* terlebih dahulu

dengan menggunakan model arsitektur *Resnext 50 dataset* 700 gambar yang sudah dilabeli, kemudian melakukan perbandingan yang akan terjadi saat melakukan *training* serta pengujian.

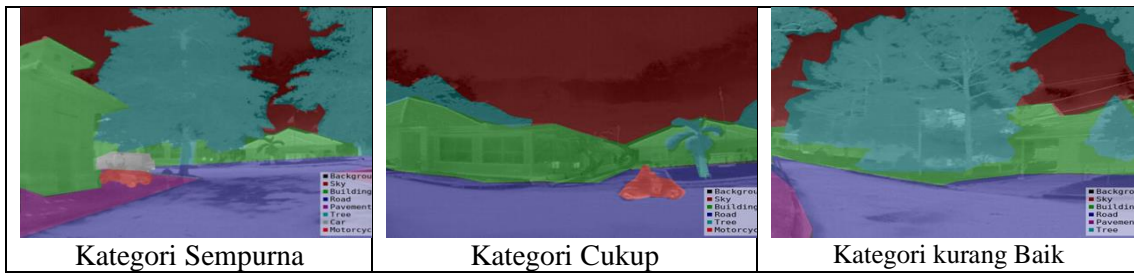
Tabel 5 Data Hasil Pengujian

| Model Segmentasi | Mean IoU (%) | Akurasi | Waktu (Hari) | Epochs |
|------------------|--------------|---------|--------------|--------|
| Resnext 50       | 75.636%      | 98.403% | 2 Hari       | 100    |

**B. Pengujian Pertama**

Pada pengujian pertama tugas akhir ini yaitu pengujian saat malam hari sistem segmentasi objek berbasis Gambar *thermal* menggunakan *Deep learning Resnext 50* untuk mensegmentasi objek disekitar ruas jalan dan lingkungan universitas Nurtanio bandung. Pengujian yang dilakukan berdasarkan nilai keakuratan serta ketepatan saat mensegmentasi objek *dataset* yang telah di *training* dengan jumlah 34030 *frame* kemudian diuji secara *offline* pada malam hari. Pada tabel dibawah ini dapat dilihat objek yang tertangkap oleh kamera *thermal*

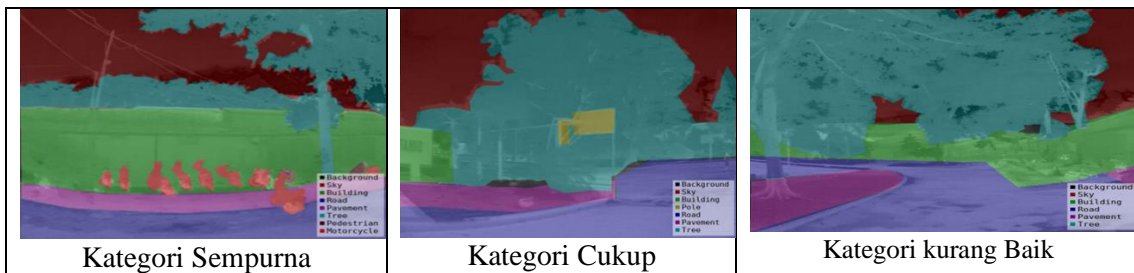
Table 6 Hasil Pengujian Pertama



**C. Pengujian Kedua**

Pada tahap ini penulis melakukan pengujian kedua sebagai perbandingan yang dimana pada pengujian ini penulis melakukan pengujian segmentasi objek berbasis gambar *thermal* saat siang hari, untuk pengujian ini menggunakan 34030 *frame* gambar diambil malam hari yang akan diuji saat siang hari, pengujian ini dilakukan berdasarkan nilai keakuratan serta ketepatan saat mensegmentasi objek baik secara *offline* maupun *realtime*.

Table 7 Hasil Pengujian Kedua



**D. Analisa Hasil**

Pengukuran hasil berdasarakan akurasi dan persentase *Intersection over Union* menjadi tolak ukur utama dalam pengujian menggunakan 34030 *frame* gambar, hal ini dimaksudkan untuk mengetahui kinerja sistem pada model arsitektur ResNext 50 saat melakukan pengujian pada gambar secara *offline*. Pada Tabel 8 akan menunjukkan data pengukuran menggunakan model arsitektur ResNext 50 yang telah dilatih dan diuji cobakan pada 34030 *frame* gambar yang diambil pada malam hari.

Tabel 8 Data Pengukuran ResNext 50

| No | Nomor <i>Frame</i> | Baik | Cukup | Kurang Baik |
|----|--------------------|------|-------|-------------|
| 1  | 0-10000            | 5021 | 3005  | 1974        |
| 2  | 10001-20000        | 5325 | 3152  | 1523        |
| 3  | 20001-34030        | 8230 | 3176  | 2594        |

$$\text{Presentasi} = \frac{\text{Jumlah Gambar}}{\sum \text{Jumlah Gambar}} \times 100\%$$

Tabel 9 Nilai Persentase Uji Fungsi ResNext 50 Pada 34030 *Frame* Gambar

| No | Kategori    | Jumlah Gambar | Persentase                                   |
|----|-------------|---------------|--|
| 1  | Baik        | 18576         | $\frac{18576}{34030} \times 100\% = 54,66\%$ |
| 2  | Cukup       | 9333          | $\frac{9333}{34030} \times 100\% = 34,43\%$  |
| 3  | Kurang Baik | 6093          | $\frac{6093}{34030} \times 100\% = 17,90\%$  |

## V. KESIMPULAN

Segmentasi objek berhasil dengan membuat label classes yang dibuat dengan format txt yang selanjutnya di *labeling* pada *LabelMe*. Dan *training* model arsitektur ResNext 50 menghasilkan 75.636% *mean Intersection over Union* (IoU) dan 98.403% akurasi. Kinerja pada model arsitektur ResNext 50 terdapat 34030 frame gambar 360° secara offline menghasilkan 18576 frame dengan kategori baik, 9333 frame kategori cukup, dan 6093 frame kategori kurang baik.

Sistem segmentasi objek menggunakan metode arsitektur FCN Resnext 50 ini dapat dikembangkan dengan menggunakan lebih banyak dataset agar dapat meningkatkan Intersection over Union pada setiap objek yang ada. Mengambil dataset pada lingkungan yang berbeda agar model software yang akan dilatih dapat memiliki banyak data untuk pembelajaran. Dapat menggunakan model arsitektur lainnya seperti ResNext 101.

## UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada pihak-pihak yang telah berkontribusi dalam penelitian ini hingga makalah ini dapat dipublikasikan pada Seminar Nasional Sains Teknologi dan Inovasi Indonesia 2021, diantaranya kepada:

1. LPDP (Lembaga Pengelola Dana Pendidikan) sebagai sponsor penelitian.
2. BRIN (Badan Riset dan Inovasi Nasional) sebagai penyedia prasarana penelitian.
3. Universitas Nurtanio sebagai penyedia prasarana penelitian.
4. AAU (Akademi Angkatan Udara) sebagai penyelenggarakonferensi dan prosiding.
5. Serta rekan-rekan peneliti

## REFERENSI

- [1] Darenoh, N. V., Prasetyo, Y., Alfarakh, M. G., & Sari, Y. A. (2014). SEGMENTASI SEMANTIK CITRA DENGAN CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN ARSITEKTUR U-NET.
- [2] Pratama, A. B., Kadir, R. E. A., & Hady, M. A. (2022). Deteksi Ruang Kosong pada Jalan Menggunakan Semantic Segmentation pada Mobil Otonom. *Jurnal Teknik ITS*, 11(1), A23-A28.
- [3] Manajang, D., Sompie, S. R. U. A., & Jacobus, A. (2021). Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor. *Jurnal Teknik Informatika*, 15(3), 171-178.
- [4] Wantania, B. B., Sompie, S. R., & Kambey, F. D. (2020). Penerapan Pendeteksian Manusia Dan Objek Dalam Keranjang Belanja Pada Antrian Di Kasir. *Jurnal Teknik Informatika*, 15(2), 101-108.

- 
- [5] Golovanov, S., Kurbanov, R., Artamonov, A., Davydov, A., & Nikolenko, S. (2018). Building detection from satellite imagery using a composite loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 229-232).
- [6] Dahria, M. (2008). Kecerdasan Buatan (Artificial Intelligence). *Jurnal Saintikom*, 5(2), 185-197.
- [7] Samuel, A. L. (1988). Some studies in machine learning using the game of checkers. II—recent progress. *Computer Games I*, 366-400.
- [8] “Fully Connected Layers in Convolutional Neural Networks” <https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/>
- [9] “Apa itu Convolutional Neural Networks?” <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- [10] Caroline dkk. (2019). *Identifikasi Jalan Kampus Universitas Sriwijaya Berbasis Fully Convolutional Network*. *SURYA ENERGY*, 2(1), 353-358.
- [11] Sik-Ho Tsang. (2018). Review: ResNeXt — 1st Runner Up in ILSVRC 2016 (Image Classification). <https://towardsdatascience.com/review-resnext-1st-runner-up-of-ilsvrc-2016-image-classification-15d7f17b42ac>
- [12] “Segmentasi semantik untuk klasifikasi citra.” <https://arifiany.medium.com/segmentasi-semantik-untuk-klasifikasi-citra-a004b3906250>
- [13] NVIDIA Developer. (2018). *Jetson AGX Xavier Developer Kit*. Diakses pada 16 Januari 2022 dari, <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>
- [14] Python. (2022). *Python*. Diakses pada 16 Januari 2022 dari, <https://www.python.org/>
- [15] *LabelMe*. (2021). *LabelMe*. Diakses pada 16 Januari 2022 dari, <http://LabelMe2.csail.mit.edu/>
- [16] “FLIR ADK Thermal Vision Automotive Development Kit Teledyne FLIR” <https://www.flir.com/products/adk/>
- [17] “Intersection Over Union” <https://medium.com/bisa-ai/intersection-over-union-a8d1532899b3>
-