Sistem Persepsi Kendaraan Otonom untuk Lingkungan Lalu-lintas Campuran pada Kondisi Cuaca Buruk dan Pencahayaan Rendah

RINGKASAN DISERTASI

Ari Wibowo
NIM: 33218022
(Program Studi Doktor Teknik Elektro dan Informatika)

Institut Teknologi Bandung
Oktober 2024

Sistem Persepsi Kendaraan Otonom untuk Lingkungan Lalu-lintas Campuran pada Kondisi Cuaca Buruk dan Pencahayaan Rendah

Disertasi ini dipertahankan pada Sidang Terbuka Sekolah Pascasarjana sebagai salah satu syarat untuk memperoleh gelar Doktor Institut Teknologi Bandung

Oktober 2024

Ari Wibowo
NIM: 33218022
(Program Studi Doktor Teknik Elektro dan Informatika)

Promotor : Prof. Dr. Bambang Riyanto T.
Ko-promotor : 1. Egi M. Idris Hidayat, Ph.D.
2. Dr. Ir. Rinaldi Munir, M.T.

Institut Teknologi Bandung
Oktober 2024

Sistem Persepsi Kendaraan Otonom untuk Lingkungan Lalu-lintas Campuran pada Kondisi Cuaca Buruk dan Pencahayaan Rendah

Ari Wibowo
NIM: 33218022

1. Latar Belakang

Perkembangan teknologi kendaraan otonom telah menunjukkan kemajuan signifikan dalam beberapa tahun terakhir. Kendaraan otonom diharapkan dapat mengurangi kecelakaan lalu lintas, meningkatkan efisiensi transportasi, dan memberikan solusi mobilitas bagi mereka yang tidak dapat mengemudi. Namun, agar dapat diadopsi secara luas, kendaraan otonom harus mampu beroperasi dengan aman dan andal dalam berbagai kondisi lingkungan. Tantangan utama yang dihadapi adalah kemampuan sistem persepsi untuk mengenali dan memahami lingkungan sekitarnya dengan akurat, terutama dalam kondisi cuaca buruk dan pencahayaan rendah. Kondisi ini sering kali mempengaruhi kinerja sensor dan algoritma yang digunakan oleh kendaraan otonom.

Cuaca buruk seperti hujan lebat, salju, kabut, dan cuaca ekstrem lainnya dapat mengganggu sensor kendaraan seperti kamera, LIDAR, dan radar. Kamera mungkin tidak dapat menangkap gambar yang jelas, LIDAR bisa mendapatkan data yang kacau karena refleksi dari partikel air, dan radar mungkin terganggu oleh interferensi elektromagnetik. Kondisi ini dapat menyebabkan kesalahan dalam deteksi objek dan penentuan jarak, yang pada akhirnya dapat membahayakan keselamatan penumpang dan pengguna jalan lainnya. Oleh karena itu, penelitian dalam meningkatkan keandalan sistem persepsi kendaraan otonom dalam kondisi cuaca buruk sangat penting.

Selain cuaca buruk, pencahayaan rendah juga merupakan tantangan besar bagi kendaraan otonom. Pada malam hari atau di daerah dengan penerangan yang minim, sensor visual seperti kamera sering kali kesulitan dalam menangkap gambar yang cukup terang untuk dianalisis. Hal ini dapat mempengaruhi kemampuan kendaraan untuk mendeteksi rambu-rambu lalu lintas, pejalan kaki, dan kendaraan lain. Teknologi seperti kamera inframerah dan sensor termal telah diusulkan sebagai solusi potensial, tetapi masih diperlukan penelitian lebih lanjut untuk mengintegrasikan dan mengoptimalkan teknologi ini dalam sistem kendaraan otonom.

Lingkungan lalu lintas yang beragam (*mixed traffic*) termasuk jalan perkotaan yang padat, jalan tol, dan jalan pedesaan, juga menambah kompleksitas sistem persepsi

kendaraan otonom. Setiap jenis lingkungan memiliki karakteristik unik yang memerlukan pendekatan berbeda dalam hal pengenalan dan pengambilan keputusan. Misalnya, jalan perkotaan mungkin memiliki banyak rambu dan marka jalan, serta kehadiran pejalan kaki dan pengendara sepeda yang tidak terduga, sementara jalan tol mungkin memerlukan deteksi kendaraan dengan kecepatan tinggi dan perencanaan jalur yang cepat. Oleh karena itu, sistem persepsi kendaraan otonom harus cukup adaptif dan fleksibel untuk mengatasi berbagai tantangan ini secara efektif.

2. Tujuan dan Sasaran Penelitian

Secara garis besar tujuan dari penelitian ini adalah mengembangkan metode atau pendekatan yang efisien pada sistem persepsi kendaraan otonom menggunakan kamera. Sistem persepsi yang dikembangkan berbasis *deep learning* yang merupakan modifikasi dari model Yolo agar dapat menangani kondisi lingkungan jalan raya yang cukup ekstrem. Arti efisien dalam penelitian ini adalah seberapa akurat model yang dikembangkan mampu mengenali objek-objek di sekitar kendaraan otonom.Untuk mencapai tujuan besar tersebut, ada beberapa tujuan spesifik yang akan dicapai secara bertahap:

1. Menghasilkan metode baru dalam mendeteksi objek pada lingkungan jalan raya untuk objek-objek kecil dan objek-objek yang samar
2. Menghasilkan metode baru yang mampu mendeteksi dan menangani objek-objek yang tertutup sebagian (teroklusi)
3. Menghasilkan metode baru yang mampu menghilangkan noise ynag mengganggu model deteksi dalam mengenali objek
4. Menghasilkan metode baru yang mampu mendeteksi objek-objek di sekitar kendaraan otonom pada kondisi cuaca buruk dan pencahayaan rendah, dan
5. Melakukan evaluasi dari metode yang dihasilkan pada poin 4 dan menerapkan pada kendaraan testbed sebagai bentuk uji coba kendaraan otonom.

3. Metode Penelitian

Pada subbab ini, dijelaskan mengenai beberapa pustaka kunci yang paling menginspirasi penulis dalam mengusulkan kontribusi utama penelitian ini. Seluruh penelitian state-of-the-art yang paling terkait dengan beberapa kontribusi pada penelitian ini dapat dilihat pada Tabel 1.

Tabel 1 Daftar penelitian state-of-the-art yang terkait dengan kontribusi utama penelitian

Artikel	Lingkup Penelitian	Kontribusi
Robust Perception in Adverse Weather Conditions Using Deep Learning (Zhang dkk., 2021)	Penggunaan *deep learning* untuk meningkatkan persepsi kendaraan otonom dalam kondisi cuaca buruk	Mengembangkan model *deep learning* yang dapat mendeteksi dan mengklasifikasikan objek

	seperti hujan, salju, dan kabut.	dengan akurasi tinggi dalam kondisi cuaca buruk
Enhancing Lidar-Based Object Detection in Rainy and Foggy Weather (Lee dkk., 2020)	Peningkatan deteksi objek berbasis Lidar dalam kondisi cuaca hujan dan kabut melalui teknik pemrosesan sinyal dan machine learning.	Memperkenalkan algoritma baru untuk menyaring noise dari tetesan air dan partikel kabut, meningkatkan akurasi deteksi objek Lidar dalam kondisi cuaca buruk
Vision-Based Navigation for Autonomous Vehicles in Adverse Weather (Kim dkk., 2019)	Teknik navigasi berbasis visi untuk kendaraan otonom dalam kondisi cuaca ekstrem seperti badai salju dan hujan lebat.	Mengembangkan metode pengolahan citra yang memperbaiki visibilitas dan deteksi jalur di bawah kondisi cuaca ekstrem, serta integrasi dengan sistem navigasi kendaraan.
WeatherNet: A Deep Learning Framework for Adverse Weather Detection and Adaptation (Li dkk., 2021)	Pengembangan *framework deep learning* untuk mendeteksi dan beradaptasi dengan kondisi cuaca buruk	Mengusulkan arsitektur *deep learning* yang dapat mendeteksi perubahan kondisi cuaca dan menyesuaikan parameter sistem persepsi secara real-time
IDOD-YOLOv7 Image Dehazing YOLOv7 for Object Detection in Low-Light Foggy Traffic Environmets (Qiu dkk., 2023)	Pengembangan *framework* deteksi objek untuk kendaraan otonom pada lingkungan berkabut	Mengusulkan arsitektur *deep learning* deteksi objek menggunakan YOLOv7 dengan penambahan modul peningkatan citra iluminasi rendah yang distel parameternya

Dalam penelitian ini, model YOLO digunakan sebagai basis untuk memodifikasi dan meningkatkan kinerjanya dalam mendeteksi objek kecil, objek yang tertutup sebagian, serta objek yang terganggu noise akibat hujan, kabut, dan pencahayaan minim. Modifikasi ini dilakukan dengan menambahkan layer-layer khusus yang dirancang untuk mengatasi tantangan-tantangan tersebut. Penambahan layer ini bertujuan untuk memperbaiki resolusi spasial deteksi objek kecil dengan memanfaatkan *feature pyramid networks* (FPN) atau teknik serupa yang meningkatkan sensitivitas terhadap detail-detail kecil. Selain itu, layer tambahan juga dirancang untuk meningkatkan ketahanan model dalam kondisi cuaca buruk dengan memanfaatkan teknik augmentasi data yang mereplikasi efek noise dari

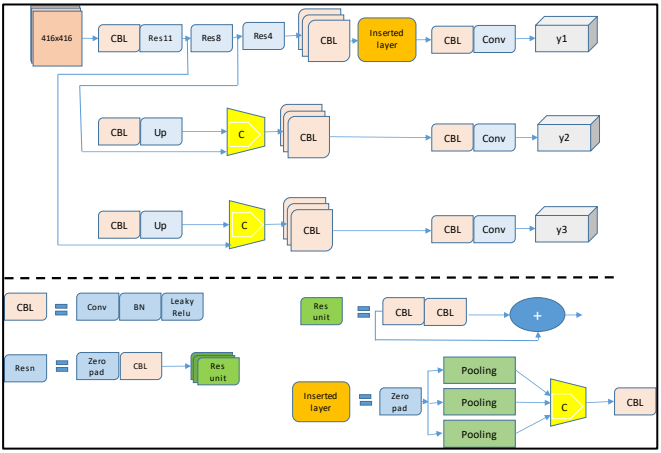
hujan, kabut, dan pencahayaan rendah, sehingga model dapat belajar mengenali objek meskipun dalam kondisi visual yang sulit.

Penelitian yang memodifikasi YOLO untuk kasus serupa telah menunjukkan bahwa adaptasi seperti ini dapat signifikan meningkatkan kinerja model dalam situasi yang menantang. Sebagai contoh, (Wu dkk., 2019) dalam penelitiannya menunjukkan bahwa dengan mengintegrasikan layer khusus untuk mendeteksi objek kecil dan objek yang tertutup sebagian, performa deteksi meningkat secara substansial. Penelitian lain oleh (Zhang dkk., 2020) menekankan pentingnya augmentasi data yang meniru kondisi cuaca buruk untuk melatih model persepsi yang lebih *robust*. Modifikasi ini tidak hanya meningkatkan akurasi deteksi tetapi juga memperkuat kemampuan model dalam situasi dunia nyata yang sering kali tidak ideal, menjadikan YOLO lebih andal dan efektif untuk aplikasi yang membutuhkan persepsi visual dalam kondisi menantang.

3.1 Modifikasi YOLOv3 dengan Penambahan *Additional Layer*

Pada subbab ini diusulkan modifikasi Yolov3 untuk mendeteksi objek yang berukuran kecil dan objek yang samar. Metode modifikasi layer yang dilakukan adalah menggabungkan penggunaan upsampling dan concatening secara bersama-sama. Arsitektur yang diusulkan ditunjukkan pada Gambar 1, terdiri dari beberapa lapisan *pooling*. *Filter* dari lapisan *pooling* ini memiliki ukuran 5x5, 9x9 dan 13x13 dengan nilai stride 1. Perubahan yang dilakukan adalah menambahkan layer untuk menggabungkan beberapa fitur pada daerah tertentu pada skala yang berbeda sehingga jaringan akan lebih tahan pada berbagai macam perubahan bentuk objek. Dengan pemilihan padding tertentu, keluaran dari layer *pooling* ini memiliki ukuran yang sama. Selanjutnya keluaran dari layer *max pooling* tersebut dikonkatenasi yang akhirnya akan menjadi masukan untuk layer konvolusi selanjutnya.

YOLOv3 memprediksi box pada 3 skala yang berbeda, yaitu 13x13, 26x26, dan 52x52 *feature map*. Sistem prediksi tersebut memiliki konsep yang sama dengan *feature pyramid networks*. Hal tersebut disebabkan hasil ekstraksi fitur yang telah melalui layer *addition*, akan di-*upsampling* sebanyak dua kali. Ketika 13x13 *feature map* digunakan untuk prediksi, di saat yang sama fitur tersebut di-*upsampling* dan dikonkatenasi dengan keluaran dari layer fitur ekstraksi menjadi 26x26. Saat fitur map 26x26 digunakan untuk prediksi, cara yang sama digunakan untuk mendapatkan fitur map 52x52.



Gambar 1 Arsitektur Yolov3Mod, Yolov3 yang dimodifikasi dengan penambahan layer

Pada arsitektur ini, layer *backbone* yang digunakan untuk ekstraksi fitur adalah darknet53. Darknet53 merupakan model fitur ekstraksi yang merupakan pengembangan dari fitur ekstraksi pada YOLOv2 yaitu darknet19. Dibandingkan dengan darknet19 yang memiliki 19-layer konvolusi, darknet53 memiliki 53-layer konvolusi. Pada dasarnya fitur ekstraksi darknet19 memiliki kesamaan dengan model fitur ekstraksi VGG, namun agak sedikit berbeda pada darknet53. Darknet53 tidak menggunakan layer *maxpooling*, melainkan menggunakan layer konvolusi secara bertahap, namun memiliki jumlah layer konvolusi yang lebih banyak. Detil dari fitur ekstraksi darknet53 dapat dilihat pada Tabel 2. Sedangkan modifikasi perubahan layer konfigurasi ditunjukkan pada Tabel 3.

Tabel 2 Detil Arsitektur *Backbone Darknet53*

	Tipe	Filter	Ukuran	Output
	Convolutional	32	3x3	256x256
	Convolutional	64	3x3/2	128x128
1x	*Convolutional*	32	1x1	
	Convolutional	64	3x3	
	Residual			128x128
	Convolutional	128	3x3/2	
2x	*Convolutional*	64	1x1	
	Convolutional	128	3x3	64x4
	Residual			32x32

	Convolutional	256	3x3/2	
8x	*Convolutional*	128	1x1	
	Convolutional	256	3x3	32x32
	Residual			16x16
	Convolutional	512	3x3/2	
8x	*Convolutional*	512	3x3	16x16
	Residual			8x8
	Convolutional	1024	3x3/2	
4x	*Convolutional*	1024	3x3	
4x	*Residual*			8x8

Tabel 3 Modifikasi penambahan dan pengurangan layer Yolov3

```
# YOLOv3-modifikasi1(pengurangan layer)

# Parameters
nc: 8
depth_multiple: 1.0
width_multiple: 1.0
anchors:
  - [10,14, 23,27, 37,58]
  - [81,82, 135,169, 344,319]

# YOLOv3-backbone
backbone:

  [[-1, 1, Conv, [16, 3, 1]],
   [-1, 1, nn.MaxPool2d, [2, 2, 0]],
   [-1, 1, Conv, [32, 3, 1]],
   [-1, 1, nn.MaxPool2d, [2, 2, 0]],
   [-1, 1, Conv, [64, 3, 1]],
   [-1, 1, nn.MaxPool2d, [2, 2, 0]],
   [-1, 1, Conv, [128, 3, 1]],
   [-1, 1, nn.MaxPool2d, [2, 2, 0]],
   [-1, 1, Conv, [256, 3, 1]],
   [-1, 1, nn.MaxPool2d, [2, 2, 0]],
   [-1, 1, Conv, [512, 3, 1]],
   [-1, 1, nn.ZeroPad2d, [[0, 1, 0, 1]]],
   [-1, 1, nn.MaxPool2d, [2, 1, 0]],
  ]

# YOLOv3-head
head:
  [[-1, 1, Conv, [1024, 3, 1]],
   [-1, 1, Conv, [256, 1, 1]],
```

```
# YOLOv3-original

# Parameters
nc: 8
depth_multiple: 1.0
width_multiple: 1.0
anchors:
  - [10,13, 16,30, 33,23]
  - [30,61, 62,45, 59,119]
  - [116,90, 156,198, 373,326]

# darknet53 backbone
backbone:

  [[-1, 1, Conv, [32, 3, 1]],
   [-1, 1, Conv, [64, 3, 2]],
   [-1, 1, Bottleneck, [64]],
   [-1, 1, Conv, [128, 3, 2]],
   [-1, 2, Bottleneck, [128]],
   [-1, 1, Conv, [256, 3, 2]],
   [-1, 8, Bottleneck, [256]],
   [-1, 1, Conv, [512, 3, 2]],
   [-1, 8, Bottleneck, [512]],
   [-1, 1, Conv, [1024, 3, 2]],
   [-1, 4, Bottleneck, [1024]],
  ]

# YOLOv3 head
head:
  [[-1, 1, Bottleneck, [1024, False]],
   [-1, 1, Conv, [512, 1, 1]],
   [-1, 1, Conv, [1024, 3, 1]],
   [-1, 1, Conv, [512, 1, 1]],
   [-1, 1, Conv, [1024, 3, 1]],
```

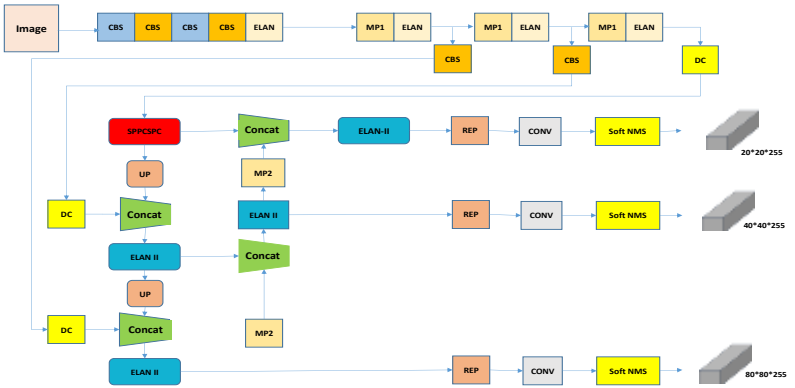
```
# YOLOv3-modifikasi2 (penambahan layer)

# Parameters
nc: 8
depth_multiple: 1.0
width_multiple: 1.0
anchors:
  - [10,13, 16,30, 33,23]
  - [30,61, 62,45, 59,119]
  - [116,90, 156,198, 373,326]

# darknet53 backbone
backbone:

  [[-1, 1, Conv, [32, 3, 1]],
   [-1, 1, Conv, [64, 3, 2]],
   [-1, 1, Bottleneck, [64]],
   [-1, 1, Conv, [128, 3, 2]],
   [-1, 2, Bottleneck, [128]],
   [-1, 1, Conv, [256, 3, 2]],
   [-1, 8, Bottleneck, [256]],
   [-1, 1, Conv, [512, 3, 2]],
   [-1, 8, Bottleneck, [512]],
   [-1, 1, Conv, [1024, 3, 2]],
   [-1, 4, Bottleneck, [1024]],
  ]

# YOLOv3-X head
head:
  [[-1, 1, Bottleneck, [1024, False]],
   [-1, 1, SPP, [512, [5, 9, 13]]],
   [-1, 1, Conv, [1024, 3, 1]],
```

[-1, 1, Conv, [512, 3, 1]], [-2, 1, Conv, [128, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 8], 1, Concat, [1]], [-1, 1, Conv, [256, 3, 1]], [[19, 15], 1, Detect, [nc, anchors]],]	[-2, 1, Conv, [256, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 8], 1, Concat, [1]], [-1, 1, Bottleneck, [512, False]], [-1, 1, Bottleneck, [512, False]], [-1, 1, Conv, [256, 1, 1]], [-1, 1, Conv, [512, 3, 1]], [-2, 1, Conv, [128, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 6], 1, Concat, [1]], [-1, 1, Bottleneck, [256, False]], [-1, 2, Bottleneck, [256, False]], [[27, 22, 15], 1, Detect, [nc, anchors]],]	[-1, 1, Conv, [512, 1, 1]], [-1, 1, Conv, [1024, 3, 1]], [-2, 1, Conv, [256, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 8], 1, Concat, [1]], [-1, 1, Bottleneck, [512, False]], [-1, 1, Bottleneck, [512, False]], [-1, 1, Conv, [256, 1, 1]], [-1, 1, Conv, [512, 3, 1]], [-2, 1, Conv, [128, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 6], 1, Concat, [1]], [-1, 1, Bottleneck, [256, False]], [-1, 2, Bottleneck, [256, False]], [-2, 1, Conv, [64, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 6], 1, Concat, [1]], [-1, 1, Bottleneck, [256, False]], [-1, 2, Bottleneck, [128, False]], [[27, 22, 15], 1, Detect, [nc, anchors]],]

3.2 Modifikasi YOLOv7 dengan Penambahan *Deformable Convolution*

Arsitektur jaringan saraf buatan yang digunakan dalam disertasi ini adalah arsitektur YOLOv7 dengan tambahan lapisan *pooling*, *deformable*, dan CBAM. Layer-layer tersebut ditambahkan setelah ekstraksi fitur yang dilakukan oleh *backbone* layer. Pada dasarnya, penambahan layer pada YOLO telah dilakukan pada penelitian (Huang dkk., 2020). Penambahan layer-layer tersebut akan meningkatkan kemampuan jaringan saraf tiruan dalam mengolah fitur yang telah diekstraksi. Ilustrasi penambahan layer dapat dilihat pada Gambar 2.

Pengembangan model YOLOV7MOD juga telah dilakukan sebelum YOLOv7MOD+M3CBAM ditunjukkan dalam Gambar 3. Pada tahap ini, tiga lapisan *deformable* embedding yang progresif ditambahkan. Yang pertama adalah menambahkan lapisan konvolusi deformasi ke kepala ekstraksi fitur pertama dari Darknet53, yang hanya memperkuat ekstraksi fitur pada peta fitur skala besar, disebut DC-1. Yang kedua adalah menambahkan lapisan konvolusi deformasi ketiga



Gambar 2 Arsitektur YOLOv7MOD

kepala ekstraksi fitur dari Darknet53, sehingga peta fitur pada setiap skala dapat mengekstraksi fitur yang lebih efektif, disebut DC-2. Yang ketiga adalah meningkatkan struktur keseluruhan Darknet53. Pertama, ukuran input jaringan disesuaikan. Ukuran citra input YOLO asli adalah 416 × 416, dan ukuran peta fitur minimum adalah 13 × 13 setelah lima kali pengurangan. Untuk meningkatkan akurasi deteksi, ukuran citra input diatur menjadi 448 × 448, dan dilakukan enam kali pengurangan, sehingga ukuran fitur minimum dari citra output adalah 7 × 7. Kemudian, lapisan konvolusi deformasi ditambahkan ke modul residual, dan kemampuan ekstraksi fitur dari konvolusi deformasi lebih kuat daripada konvolusi konvensional. Oleh karena itu, jumlah lapisan jaringan dapat disesuaikan secara tepat untuk mengurangi jumlah perhitungan. Modul residual yang ditingkatkan terdiri dari lapisan *deformable* DC-3 dan lapisan konvolusi 1 × 1, 3 × 3. Deteksi objek akhir dalam arsitektur ini menggunakan Soft-NMS. Algoritma YOLOV7 yang ditingkatkan ini disebut YOLOV7MOD.

Pada YOLOv7mod, penggunaan tiga lapisan DC setelah layer ELAN dan CBS bertujuan untuk meningkatkan kemampuan model dalam menangkap fitur spasial yang kompleks dan mengatasi variasi bentuk serta posisi objek. *Deformable* Convolution memperkenalkan *deformable* offsets yang memungkinkan konvolusi untuk beradaptasi dengan bentuk objek yang tidak teratur, meningkatkan fleksibilitas dan adaptasi model. Penempatan layer DC setelah ELAN (*Efficient Layer Aggregation Network*) dan CBS (*Convolution, Batch Normalization, Swish activation*) memungkinkan konvolusi *deformable* untuk bekerja pada fitur yang sudah diperkuat dan diperkaya, meningkatkan efisiensi komputasi dan efektivitas deteksi pada berbagai skala. Dengan tiga lapisan DC, model dapat mengintegrasikan informasi spasial dari berbagai skala, menyediakan fitur yang lebih *robust* untuk

deteksi objek kecil dan besar, serta memperbaiki representasi fitur dengan menggabungkan konteks yang lebih luas.

Gambar 3 Arsitektur YOLOv7MOD+M3CBAM

Penambahan layer M3CBAM dilakukan setelah layer fitur extraksi. Tujuan dari layer ini adalah untuk menggabungkan beberapa fitur pada daerah tertentu pada skala yang berbeda, sehingga jaringan akan lebih tahan pada berbagai macam perubahan bentuk objek. M3CBAM layer terdiri dari beberapa layer CAM dan SAM. Filter dari layer *pooling* ini memiliki ukuran 5x5, 9x9 dan 13x13 dengan nilai stride 1. Dengan pemilihan padding tertentu, keluaran dari layer *max pooling* ini memiliki ukuran yang sama. Selanjutnya keluaran dari layer *max pooling* tersebut dikonkatenasi yang akhirnya menjadi masukan untuk layer konvolusi selanjutnya. Karena beberapa ukuran fitur yang berbeda digabungkan, maka jaringan dapat menggunakan data spasial yang lebih banyak pada layer konvolusi.

YOLOv7 memprediksi bounding box pada tiga skala yang berbeda, yaitu 13x13, 26x26, dan 52x52 feature map, untuk memungkinkan deteksi objek dengan berbagai ukuran secara efektif. Sistem prediksi ini mengadopsi konsep yang mirip dengan FPN, yang memanfaatkan fitur dari berbagai skala. Setelah melalui layer M3CBAM, fitur yang diekstraksi akan di-**upsample** dua kali. Ketika feature map 13x13 digunakan untuk prediksi, pada saat yang sama, fitur tersebut di-**upsample** dan dikonkatenasi dengan keluaran dari layer ekstraksi fitur, menghasilkan feature map 26x26. Proses serupa diterapkan lagi untuk mendapatkan feature map 52x52 dari feature map 26x26. Setiap hasil **upsampling** akan masuk ke layer *deformable* untuk

ekstraksi fitur yang lebih dinamis dan adaptif. Layer *deformable* ini membantu dalam menyesuaikan fitur terhadap variasi bentuk dan posisi objek. Akhirnya, hasil dari layer *deformable* digabungkan untuk mendapatkan fitur yang lebih kaya dan lengkap, yang kemudian digunakan untuk prediksi bounding box pada masing-masing skala (13x13, 26x26, dan 52x52).

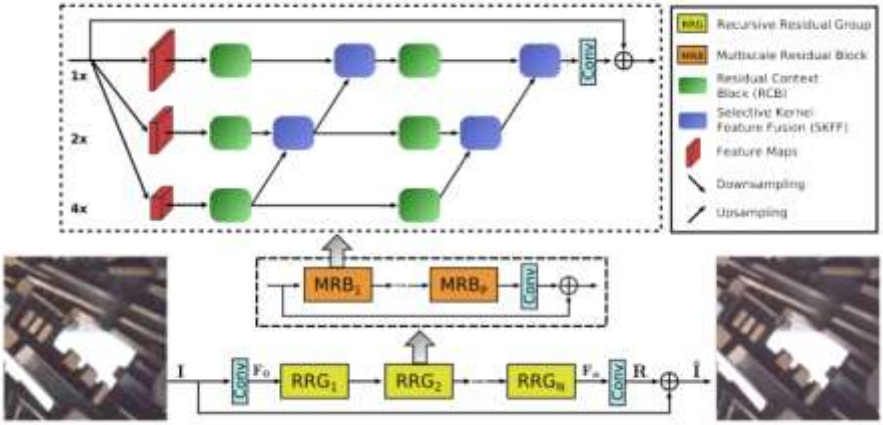
YOLOv7MOD menggunakan backbone berbasis **CSPDarknet53**, yang merupakan peningkatan dari Darknet53 dengan penambahan CSPNet (Cross Stage Partial Network). CSPDarknet pertama kali diperkenalkan dalam YOLOv4 dan telah dimodifikasi dalam YOLOv7 untuk meningkatkan efisiensi dan performa model. Selain itu, YOLOv7MOD menambahkan beberapa modifikasi seperti M3CBAM dan layer *deformable* untuk meningkatkan kemampuan model dalam menangkap fitur yang lebih kaya dan variatif. Secara rinci, CSPDarknet53 terdiri dari beberapa tahap yang diuraikan sebagai berikut:

1. ***Stage 1***:
    a. *1 x Conv (3x3, 32 filters)*
    b. *1 x Conv (3x3, 64 filters, stride 2)*
    c. *CSP Block dengan 1 x Residual Block*
2. ***Stage 2***:
    a. *1 x Conv (3x3, 128 filters, stride 2)*
    b. *CSP Block dengan 2 x Residual Blocks*
3. ***Stage 3***:
    a. *1 x Conv (3x3, 256 filters, stride 2)*
    b. *CSP Block dengan 8 x Residual Blocks*
4. ***Stage 4***:
    a. *1 x Conv (3x3, 512 filters, stride 2)*
    b. *CSP Block dengan 8 x Residual Blocks*
5. ***Stage 5***:
    a. *1 x Conv (3x3, 1024 filters, stride 2)*
    b. *CSP Block dengan 4 x Residual Blocks*

3.3 Modifikasi Model Image Denoising dengan Self Attention

Untuk melakukan proses denoising metode yang dipakai adalah MIRNet-v2. Skema MIRNet-v2 yang diusulkan ditunjukkan pada Gambar 4. Pada diagram tersebut terdiri dari lapisan konvolusi dan beberapa lapisan *recursive residual group* (RRG). Setiap lapisan RRG memuat beberapa lapisan *multiscale residual block* (MRB) yang di dalamnya memberikan rincian blok sisa multi-skala, yaitu blok bangunan mendasar dari metode yang berisi beberapa elemen kunci: (a) aliran konvolusi multi-resolusi paralel untuk mengekstraksi representasi fitur (halus hingga kasar) yang lebih kaya secara semantik dan (kasar hingga halus) yang presisi secara spasial, (b) pertukaran informasi antar multi-resolusi, (c) agregasi fitur-fitur yang datang dari

aliran yang berbeda, dan (d) blok residual contex ke pengekstraksi fitur berbasis atensi.



Gambar 4 Diagram Arsitektur Image Denoising MIRNet-v2 (Waqas dkk., 2022)

Penjelasan diagram pipeline keseluruhan sebagai berikut, diberikan gambar $I \in R^{HxWx3}$, model yang diusulkan pertama-tama menerapkan lapisan konvolusional untuk mengekstrak fitur tingkat rendah $Fo \in R^{HxWxC}$. Selanjutnya, peta fitur Fo melewati N jumlah lapisan residual recursive group (RRG), menghasilkan fitur dalam $Fn \in R^{HxWxC}$. Pada setiap lapisan RRG berisi beberapa blok multi-scale residual, yang dijelaskan lebih detil pada bagian III.2. Selanjutnya menerapkan lapisan konvolusi ke fitur dalam Fn dan mendapatkan citra sisa $R \in R^{HxWx3}$. Akhirnya, citra yang dipulihkan diperoleh sebagai $\hat{I} = I + R$. Untuk mengoptimalkan jaringan yang digunakan fungsi loss Charbonnier [6]:

$$L(\hat{I}, I^{*}) = \sqrt{|\hat{I} - I*|^2 + \varepsilon^2} \quad (1)$$

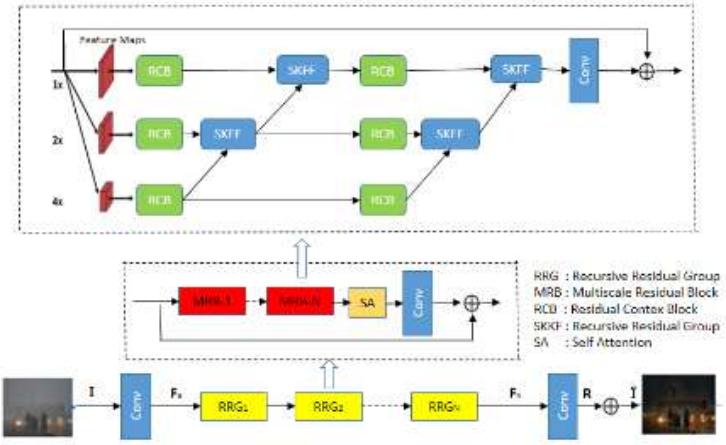
dimana I^{*} adalah ground-truth dan ε adalah konstanta yang nilainya 10^{-3}.

Pada disertasi ini juga diusulkan model denoising gambar MIRSA yang merupakan modifikasi dari model MIRNet-v2 dengan mengintegrasikan mekanisme *self attention* (SA) untuk meningkatkan kinerja.

Diagram skema denoising gambar MIRSA yang sudah dikembangkan ditunjukkan pada Gambar 5. SA diwakili oleh simbol matematika. Simbol SA yang paling umum digunakan adalah SelfAttention(·) atau SA(·). Jika mengambil representasi vektor atau matriks sebagai masukan, maka dapat menggunakan operasi SA dengan simbol-simbol seperti berikut:

$$\mathbf{SA}(\boldsymbol{X}) = \mathbf{Softmax}\left(\frac{\boldsymbol{X}.\boldsymbol{X}'}{\boldsymbol{sqrt}(\boldsymbol{d})}\right).\boldsymbol{X} \quad (2)$$

Di sini, X adalah masukan, · menunjukkan operasi perkalian matriks, X' adalah transpos X, d adalah faktor skala, dan softmax(·) adalah fungsi softmax yang diterapkan pada setiap baris matriks untuk memperoleh bobot. SA dalam denoising MIRSA adalah mekanisme yang memungkinkan model untuk secara adaptif menekankan bagian-bagian penting dari gambar masukan sekaligus mengurangi noise. SA diimplementasikan untuk meningkatkan pemahaman model terhadap hubungan spasial antar piksel pada gambar. Cara kerja SA dimulai dengan menghasilkan tiga proyeksi linier dari representasi fitur masukan yaitu kunci (W_k), nilai (W_v), dan pertanyaan (W_q). Representasi ini kemudian digunakan untuk menghitung skor perhatian yang mengukur sejauh mana kontribusi setiap piksel terhadap setiap piksel lainnya. Skor perhatian ini dihitung dengan memproyeksikan pertanyaan dan kunci, kemudian menerapkan fungsi softmax untuk mendapatkan bobot perhatian. Bobot perhatian ini memberikan arti penting pada setiap posisi dalam gambar, menekankan piksel yang lebih relevan dalam pemrosesan. Setelah memperoleh bobot perhatian, nilai-nilai yang sesuai digunakan untuk menghasilkan representasi hasil dengan memanfaatkan informasi dari keseluruhan gambar dengan proporsi bobot yang sesuai. Intinya, SA memungkinkan MIRSA memutuskan sejauh mana setiap piksel harus memperhatikan informasi dari piksel lain, sehingga meningkatkan kemampuan model untuk menangkap struktur dan pola kompleks dalam gambar. SA memungkinkan model untuk fokus pada area di sekitar noise atau tepi objek penting, sambil mengabaikan bagian yang kurang relevan. Dengan cara ini, model dapat menghilangkan noise dengan lebih efektif tanpa mengorbankan detail penting pada gambar. Selain itu, membantu mengatasi tantangan dalam melakukan denoising, seperti mempertahankan tekstur halus dan mengidentifikasi fitur yang kurang jelas karena noise.



Gambar 5 Diagram Arsitektur Image Denoising MIRSA

Berikut adalah beberapa poin yang menjelaskan mengapa SA diletakkan setelah layer MRB:

1. Pemrosesan Fitur yang Lebih Efektif: MRB digunakan untuk menangkap dan memperkuat fitur multi-skala dari gambar yang berisik. Setelah fitur-fitur ini diproses melalui MRB, mereka menjadi lebih representatif dan kaya informasi. Menempatkan SA setelah MRB memungkinkan mekanisme SA untuk bekerja dengan fitur-fitur yang sudah diperkaya ini, sehingga dapat lebih efektif dalam memahami hubungan panjang antara piksel dalam gambar.
2. Optimalisasi Kapasitas Jaringan: SA memerlukan komputasi yang tinggi, terutama dalam gambar dengan resolusi tinggi. Dengan menempatkannya setelah MRB, jumlah fitur yang harus diproses oleh SA dapat dikurangi melalui proses downsampling yang dilakukan oleh MRB. Hal ini membantu dalam mengurangi beban komputasi dan penggunaan memori, membuat jaringan lebih efisien.
3. Pemahaman Konteks yang Lebih Baik: Setelah MRB, fitur-fitur gambar memiliki konteks multi-skala yang lebih kaya. SA yang bekerja pada fitur-fitur ini dapat mengidentifikasi dan memodelkan hubungan yang lebih kompleks antara berbagai bagian gambar.
4. Stabilitas dan Kinerja Pelatihan: Menempatkan SA setelah layer MRB juga dapat membantu dalam stabilitas pelatihan. MRB dapat bertindak sebagai mekanisme regularisasi yang mempersiapkan fitur-fitur untuk diproses oleh SA, sehingga membantu menghindari masalah pelatihan yang mungkin timbul dari penggunaan SA secara langsung pada fitur-fitur yang belum diproses.

3.4 Peningkatan Kinerja MIRSA+YOLOv7MOD dengan Multiscale CBAM

Pertimbangan dalam memilih MIRNet-v2 sebagai komponen utama dalam pengembangan deteksi objek menggunakan YOLOV7MOD dapat didasarkan pada beberapa faktor kunci, antara lain Kemampuan Denoising MIRNet-v2 dan Peningkatan Akurasi Deteksi.

Kesesuaian antara arsitektur MIRNet-v2 dan YOLOV7MOD perlu dipertimbangkan. Kombinasi kedua model ini diharapkan dapat memberikan sinergi yang kuat, memungkinkan YOLOV7MOD untuk lebih baik menangani tantangan deteksi objek, terutama dalam konteks lingkungan dengan noise visual yang tinggi. Ketersediaan dukungan pengembang dan dokumentasi yang baik untuk MIRNet-v2 adalah faktor penting dalam memastikan integrasi yang lancar dan mempermudah proses pengembangan dan penyesuaian. Seperti sudah dijelaskan pada Bab II.1, bahwa model denoising yang digunakan telah dilakukan modifikasi dengan penambahan self attention. Model baru yang dihasilkan dinamakan MIRSA, selanjutnya istilah ini yang akan digunakan pada saat melakukan penggabungan model denoising dengan model objek deteksi.

Proses penggabungan kedua model dapat dijelaskan pada paragrap ini. Model menerima citra input dengan resolusi tertentu, dan input ini melewati serangkaian lapisan konvolusi di dalam backbone untuk menghasilkan fitur-fitur yang diperlukan untuk deteksi objek. Bagian yang paling sesuai untuk menerima input dari MIRSA adalah input layer dan backbone. Input layer dapat menerima citra yang telah melalui proses denoising oleh MIRSA, sementara Backbone memproses fitur-fitur dari citra tersebut. Output dari MIRSA, yang telah menjalani denoising, dapat diintegrasikan pada tahap input layer YOLOV7MOD. Hal ini memastikan bahwa citra yang diterima oleh YOLOV7MOD sudah bersih dari noise, meningkatkan kemampuan deteksi pada objek teroklusi atau citra yang terpengaruh oleh noise visual. Detection head pada YOLOV7MOD terdiri dari beberapa lapisan konvolusi dan lapisan deteksi yang menghasilkan bounding box dan skor kepercayaan untuk setiap kelas objek. MIRSA dapat diintegrasikan dengan head ini untuk memberikan input yang telah melalui proses denoising. YOLOV7MOD menggunakan Darknet-53 sebagai backbone, yang terdiri dari 53 layer konvolusi. Ini bertanggung jawab untuk mengekstrak fitur-fitur tingkat tinggi dari citra input. Berikutnya adalah menyesuaikan ukuran dan format output dari MIRSA agar sesuai dengan input yang diperlukan oleh YOLOV7MOD. Pada tahap ini harus dipastikan bahwa output dari MIRSA dapat digunakan sebagai input untuk bagian deteksi objek. Integrasikan output dari MIRSA sebagai lapisan tambahan pada backbone YOLOV7MOD. Ini dapat dilakukan melalui operasi konkatensi atau penggabungan yang sesuai dengan struktur arsitektur. Lakukan fine-tuning pada model gabungan untuk memastikan bahwa interaksi antara MIRSA dan YOLOV7MOD berjalan dengan baik. Ini melibatkan penyesuaian parameter dan pembelajaran ulang untuk memastikan bahwa model bersatu memiliki kinerja yang optimal.

Arsitektur MIRSA+YOLOv7MOD yang merupakan gabungan dari image denoising dan objek deteksi dapat ditunjukkan pada Gambar 6 di bawah ini.

Gambar 6 Arsitektur Model Deteksi dan Denosing MIRSA+YOLOV7MOD

Pada arsitektur di atas sistem gabungan antara MIRSA untuk denoising citra dan pendeteksi objek YOLOV7MOD pada sistem persepsi kendaraan otonom dirancang untuk meningkatkan kemampuan penglihatan kendaraan otonom di lingkungan yang bising atau berisik. Pada arsitektur ini blok denoising MIRSA mengacu pada blok

diagram pada Gambar 5, sedangkan blok deteksi YOLOv7MOD mengacu pada Gambar 2.

Berikut adalah penjelasan rinci tentang cara kerja sistem gabungan ini:

1. Image Denoising menggunakan MIRSA:
   a. Input Citra: Citra yang diambil dari sensor kendaraan otonom masuk ke MIRSA sebagai input.
   b. Pembersihan Noise: MIRSA bertanggung jawab untuk membersihkan noise atau gangguan pada citra tersebut. MIRSA menggunakan struktur jaringan residual yang memungkinkan pembelajaran dari instance ganda, sehingga dapat mengatasi noise secara efektif.
   c. Denoised Image Output: Hasil dari MIRSA adalah citra yang telah dihilangkan noise-nya, sehingga lebih baik dalam memberikan representasi visual yang bersih dan jelas.

2. Pendeteksian Objek menggunakan YOLOV7MOD:
   a. Input Citra yang Telah Dibenahi: Citra yang telah dibersihkan dari noise oleh MIRSA menjadi input untuk YOLOV7MOD.
   b. Ekstraksi Fitur: YOLOV7MOD menggunakan arsitektur CSPDarknet53 sebagai tulang punggung (backbone) untuk mengekstraksi fitur-fitur penting dari citra. Ini membantu dalam menangkap konteks dan detail objek.
   c. *Deformable Convolution* (DC): YOLOV7MOD memanfaatkan lapisan DC secara strategis, lapisan ini memungkinkan jaringan untuk menyesuaikan grid sampling secara dinamis, memperbaiki kekurangan konvolusi tradisional terutama dalam mendeteksi objek yang sebagian terhalang atau memiliki bentuk yang tidak teratur.
   d. Deteksi Objek: proses ini menghasilkan kotak pembatas (bounding box) dan label objek pada citra.

3. Gabungan dan Output Sistem Persepsi:
   a. Gabungan Output: Hasil dari MIRSA dan YOLOV7MOD digabungkan untuk menghasilkan citra yang telah dibersihkan dari noise dan dilengkapi dengan deteksi objek.
   b. Output Sistem Persepsi: Citra akhir ini, yang telah melalui proses denoising dan deteksi objek, dianggap sebagai output dari sistem persepsi kendaraan otonom. Ini memberikan informasi visual yang lebih bersih dan kontekstual, yang penting untuk keputusan dan tindakan yang diambil oleh sistem otonom dalam pengaturan lalu lintas dan navigasi.

Sistem ini memberikan perbaikan signifikan dalam persepsi kendaraan otonom di lingkungan yang mungkin penuh dengan gangguan atau noise, serta meningkatkan

kemampuan deteksi objek, terutama untuk objek yang sebagian terhalang atau memiliki bentuk yang tidak teratur.

Gambar 7 Arsitektur Model Deteksi dan Denoising MIRSA+YOLO7MOD+M3CBAM

Arsitektur gabungan denoising dan deteksi objek dengan optimasi multiscale CBAM pada gambar 7 adalah suatu pendekatan yang menggabungkan dua tugas utama dalam pengolahan citra, yaitu membersihkan gambar dari noise (gangguan) dan mendeteksi objek dalam gambar tersebut, dengan menggunakan optimasi multiscale CBAM untuk meningkatkan kinerja sistem. Pada arsitektur ini blok denoising MIRSA mengacu pada blok diagram pada Gambar 5, sedangkan blok deteksi YOLOv7MOD+M3CBAM mengacu pada Gambar 3.

Berikut adalah uraian penjelasan tentang arsitektur ini:

1. Preprocessing Denoising (Pembersihan Noise):
   Langkah pertama dalam arsitektur ini adalah membersihkan gambar input (gambar X) dari gangguan atau noise menggunakan teknik denoising.
   Teknik denoising yang digunakan dapat bervariasi, mulai dari teknik dasar seperti *filter median* atau *filter gaussian*, hingga teknik yang lebih canggih seperti penggunaan jaringan saraf tiruan khusus untuk denoising.
2. Ekstraksi Fitur Multiscale:
   Setelah membersihkan gambar dari noise, langkah berikutnya adalah mengekstrak fitur-fitur penting dari gambar menggunakan jaringan saraf konvolusi (CNN) multiscale. Jaringan CNN multiscale menghasilkan representasi fitur pada berbagai tingkat resolusi atau skala, memungkinkan sistem untuk menangkap detail-detail penting dari gambar dalam berbagai tingkat kompleksitas.
3. Optimasi Multiscale CBAM:
   Setelah ekstraksi fitur multiscale, fitur-fitur yang dihasilkan kemudian diperlakukan oleh modul perhatian multiscale CBAM.
   Modul perhatian multiscale CBAM menyoroti fitur-fitur penting pada berbagai skala dan tingkat abstraksi, membantu meningkatkan kemampuan sistem dalam mendeteksi objek dengan berbagai ukuran dan kompleksitas.
4. Deteksi Objek:

Setelah proses optimasi dengan multiscale CBAM, langkah terakhir adalah mendeteksi objek dalam gambar menggunakan detektor objek yang sesuai. Detektor objek dapat berupa jaringan deteksi objek seperti YOLO (You Only Look Once) atau SSD (Single Shot MultiBox Detector), yang telah diatur untuk mendeteksi objek pada berbagai skala dan lokasi dalam gambar.

5. Post-processing:
   Setelah deteksi objek dilakukan, gambar hasil dapat melalui tahap post-processing untuk meningkatkan akurasi dan keandalan deteksi.
   Post-processing dapat mencakup langkah-langkah seperti *filtering* deteksi ganda, pemilihan kotak pembatas (bounding box) yang paling sesuai, atau penggabungan deteksi yang saling tumpang tindih.

Pendekatan yang menggabungkan denoising, ekstraksi fitur multiscale, dan deteksi objek dengan optimasi multiscale CBAM, arsitektur ini memiliki potensi untuk meningkatkan kinerja sistem dalam mengenali dan mengekstrak objek dari gambar yang terkena gangguan noise, serta meningkatkan ketepatan deteksi objek pada berbagai skala dan kondisi lingkungan.

4. Hasil dan Pembahasan

Pada bab ini, ditampilkan hasil-hasil percobaan dengan simulasi terkait dengan algoritma yang diusulkan. Selanjutnya, analisis dari beberapa hasil yang didapatkan dengan beberapa skenario yang berbeda dilakukan untuk menunjukkan kelebihan dan kekurangan dari metode yang diusulkan.

4.1 Model YOLOv3MOD

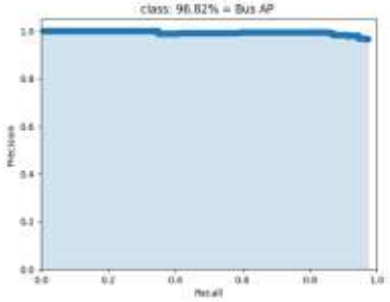
Berikut beberapa skenario pengujian yang akan diujicobakan pada Yolov3Mod yaitu objek jelas, objek kecil, dan objek samar. Dataset deteksi objek terdiri dari 14132 citra dengan delapan kelas yang berbeda, yaitu Pengendara Sepeda Motor, Pedestrian, Mobil, Truk, Bus, Mini Truck Box, Mini Pickup, dan Minibus. Dataset dibagi untuk pelatihan, pelatihan validasi, dan pengujian. Jumlah citra yang digunakan untuk pelatihan sebanyak 12719 citra, untuk pelatihan validasi sebanyak 1413 citra untuk keperluan pengujian sebanyak 1585 citra. Beberapa metode pembelajaran diuji untuk melatih model YOLOv3 dan YOLO-addition untuk mencapai metode pelatihan terbaik yang menghasilkan angka presisi tertinggi dengan dataset terkini dan epoch minimum. Setelah model dilatih dengan dataset, model diuji dengan dataset testing dengan IOU threshold 0.5 dan confidence score 0.1. Didapatkan hasil seperti pada Tabel 4 di bawah ini.

Tabel 4 Perbandingan hasil kualitatif pengujian Yolov3 dengan Yolov3Mod

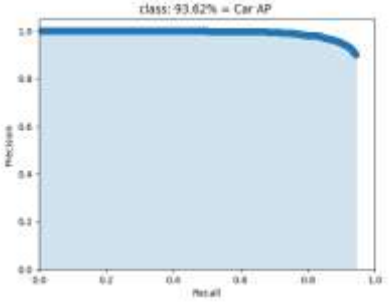
No	Yolov3 ori	Yolov3 mod
1		
2		
3		
4		

4.2 Model YOLOv7MOD

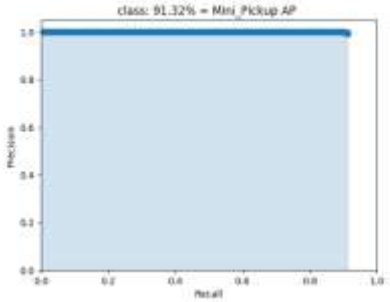
Pada metode keenam, digunakan training dengan dua tahap yaitu *transfer learning* dan dilanjutkan dengan *finetuning* pada model YOLOv7-MOD. Dengan metode ini, *pre-trained backbone* layer yaitu darknet53 diimplementasikan saat training. Sehingga model yang digunakan telah memiliki *backbone* dengan bobot yang telah dilatih dengan dataset MS COCO. Model ini dilatih dengan jumlah 200 periode. Namun berbeda dengan sebelumnya, terdapat dua tahapan dalam training ini. Dalam 20 periode pertama *backbone* layer akan dibekukan, dan 180 periode selanjutnya *backbone* layer dapat di-*finetuning*. *Learning rate* diinisialisasi dengan nilai 0.001 dan akan berkurang dengan factor pengalian 0.1 apabila setiap 8 periode berturut-turut nilai *validation loss* tidak berkurang. Selain itu, apabila dalam 20 periode nilai *validation loss* tidak berkurang, maka training akan dihentikan. Dari hasil training tersebut, diperoleh hasil kurva *precision-recall* dibawah ini.


class: 98.82% = Bus AP
Precision
Recall

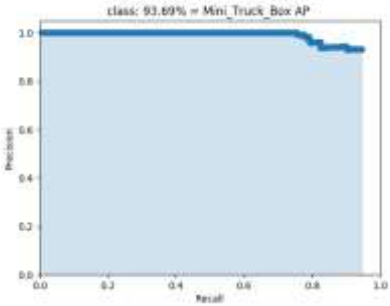
a.


class: 93.62% = Car AP
Precision
Recall

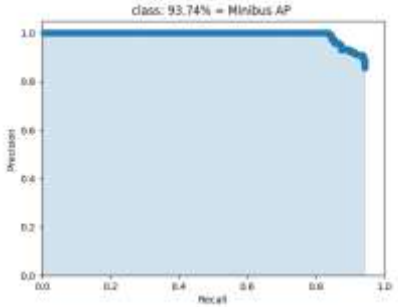
b.


class: 91.32% = Mini_Pickup AP
Precision
Recall

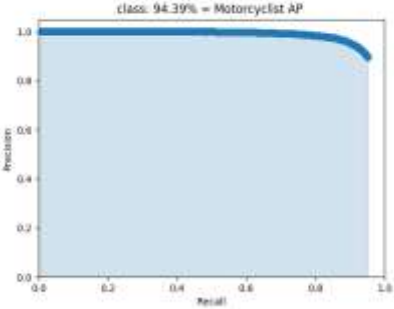
c.


class: 93.69% = Mini_Truck_Box AP
Precision
Recall

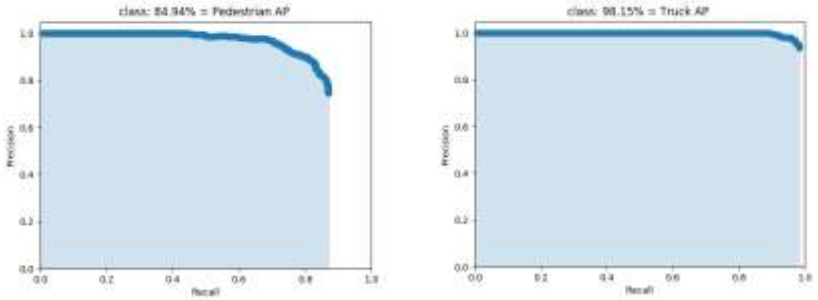
d.


class: 93.74% = Minibus AP
Precision
Recall

e.


class: 94.39% = Motorcyclist AP
Precision
Recall

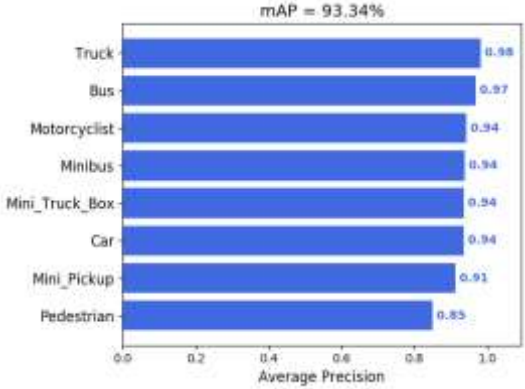
f.



g. h.

Gambar 8 Precision-recall untuk YOLOv7-MOD dengan transfer learning dengan finetune learning untuk a. Bus, b. Car, c. Mini Pickup, d. Mini Truck Box, e. Minibus, f. Motorcyclist, g. Pedestrian, h. Truck

Penggunaaan metode *transfer learning* dengan *finetuning* pada YOLOv7-MOD menghasilkan kurva *precision-recall* yang cukup baik. Dapat dilihat pada kurva *precision-recall*, Pedestrian adalah kelas yang paling sering terjadi kesalahan pendeteksian mencapai nilai AP sebesar 84.94%. Selain itu, YOLOv7-MOD dengan *transfer learning* dengan *finetuning* masih memiliki kelas dengan nilai presisi yang rendah, Mini Truck Box memiliki presisi dengan nilai 91.32%. Sehingga, secara keseluruhan hasil yang diperoleh dari model ini masih dibawah YOLOv7-MOD dengan *transfer learning* dengan *backbone* yang dibekukan. Setelah semua hasil AP didapat, nilai tersebut dirata rata sehingga diperoleh hasil mAP sebesar 93.34%.



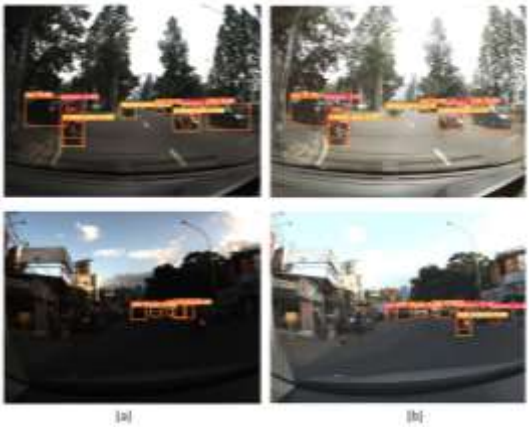
Gambar 9 Nilai AP setiap kelas dan hasil akhir mAP dari YOLOv7-MOD dengan Transfer learning dan Finetuning

4.3 Model MIRSA+YOLOv7MODM3CBAM

Untuk memvalidasi kinerja model optimasi gabungan MIRSA+YOLOV7MOD yang diusulkan dalam riset ini untuk deteksi objek di lingkungan dengan cahaya rendah, dilakukan uji cross-sectional. dan eksperimen perbandingan dilakukan pada kumpulan data pengujian yang sama. Pertama, algoritma MIRSA+YOLOV7MOD dibandingkan dengan detektor objek menggunakan CNN yang canggih saat ini antara lain metode Fast R-CNN, Faster R-CNN, SSD, RetinaNet, dan YOLOV7. Tabel II.3 menunjukkan hasil deteksi detektor yang berbeda pada konsentrasi gambar kabut yang berbeda. Terlihat pada tabel, akurasi pendeteksian objek menggunakan metode MIRSA+YOLOV7MOD lebih baik dibandingkan algoritma pendeteksian di atas pada kondisi minim cahaya.

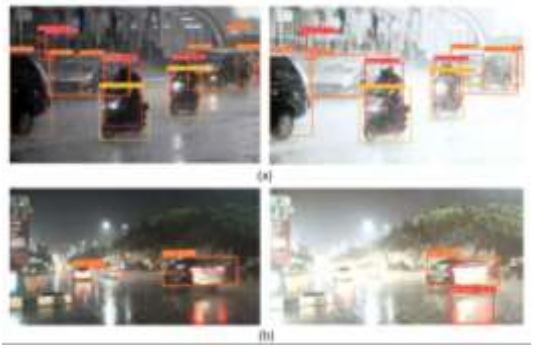
a. Kondisi sore menjelang malam.

   Pengujian ini dilakukan untuk mengetahui pengaruh pencahayaan yang minim terhadap akurasi model deteksi yang dihasilkan. Pada Gambar 10 ditunjukkan bahwa untuk hasil deteksi pada citra asli (a) ada beberapa objek yang tidak dikenali berupa penanda kelas yaitu *bounding box*. Sedangkan pada citra yang sudah dihilangkan *noise*-nya (b) beberapa objek dapat dikenali kembali. Hal ini menunjukkan secara kualitatif model deteksi mengalami peningkatan performansi jika dibandingkan dengan model yang dikembangkan.



Gambar 10 Hasil Deteksi Objek pada Kondisi Sore Menjelang Malam

b. Kondisi hujan



MIRSA+YOLOV7MOD

MIRSA+YOLOV7MODM3CBAM

Gambar 11 Perbandingan Kualitatif Hasil Deteksi Objek pada Kondisi Hujan, (a) hujan pada siang hari, (b) hujan pada malam hari

Tabel 5 Pengujian MIRNet-v2 pada Dataset LLD dengan Variasi Ukuran Blok

Jumlah Layer RRG	Jumlah Layer MRB			
	1	2	3	4
1	15.67	18.55	27.81	29.48
2	19.32	25.56	31.53	37.79
3	25.59	32.09	34.78	37.86
4	27.18	**38.06**	38.45	40.35

Pada Tabel 5 disajikan pengujian MIRNet-v2 menggunakan metrik PSNR dengan kombinasi jumlah layer RRG dan MRB menunjukkan hasil yang menarik. Meskipun nilai tertinggi PSNR tercapai pada konfigurasi dengan 4 RRG dan 4 MRB serta 3 RRG dan 4 MRB, perlu dicatat bahwa waktu komputasi yang sangat lama dapat menjadi kendala. Waktu komputasi yang lambat dapat merugikan efisiensi penggunaan model, terutama dalam skenario yang memerlukan responsibilitas waktu nyata. Oleh karena itu, pengoptimalan jumlah layer menjadi 4 RRG dan 2 MRB, yang memberikan keseimbangan antara hasil denoising yang baik dan waktu komputasi yang lebih cepat, merupakan pilihan yang masuk akal. Ini menunjukkan bahwa dalam konteks penggunaan praktis, seringkali perlu adanya penyesuaian kompromi antara kinerja dan efisiensi, terutama dalam hal waktu komputasi yang sesuai dengan kebutuhan aplikasi sehari-hari.

Tabel 6 Evaluasi kuantitatif pada kumpulan data denoising

Metode	PSNR	SSIM
RIDNet	28.23	0.628
DnCNN	30.25	0.790
MLP	32.05	0.833
BM3D	32.33	0.851
SADNet	37.39	0.952
DANet	37.38	0.955
CycleISP	37.46	0.956
MIRNet-v2	37.79	0.956
MIRSA	**38.06**	**0.958**

Pada Tabel 6 ditampilkan perbandingan antar model denoising untuk digabungkan dengan model deteksi, dari hasil evaluasi model denoising MIRSA memiliki nilai PSNR dan SSIM tertinggi dibanding model yang lain, yaitu 38.06 dan 0.958. Evaluasi ini dilakukan pada kumpulan dataset LLD yang sudah diambil diambil sebelumnya pada lingkungan mixed traffic dan pencahayaan kurang.

Tabel 7 Perbandingan mAP pada beberapa metode deteksi objek

Metode	Berat	Sedang	Ringan
Fast R-CNN [30]	0.6237	0.6943	0.7972
Faster R-CNN [31]	0.6416	0.7317	0.8118
SSD [32]	0.6434	0.7347	0.8218
YOLOV3 [33]	0.6148	0.7016	0.7764

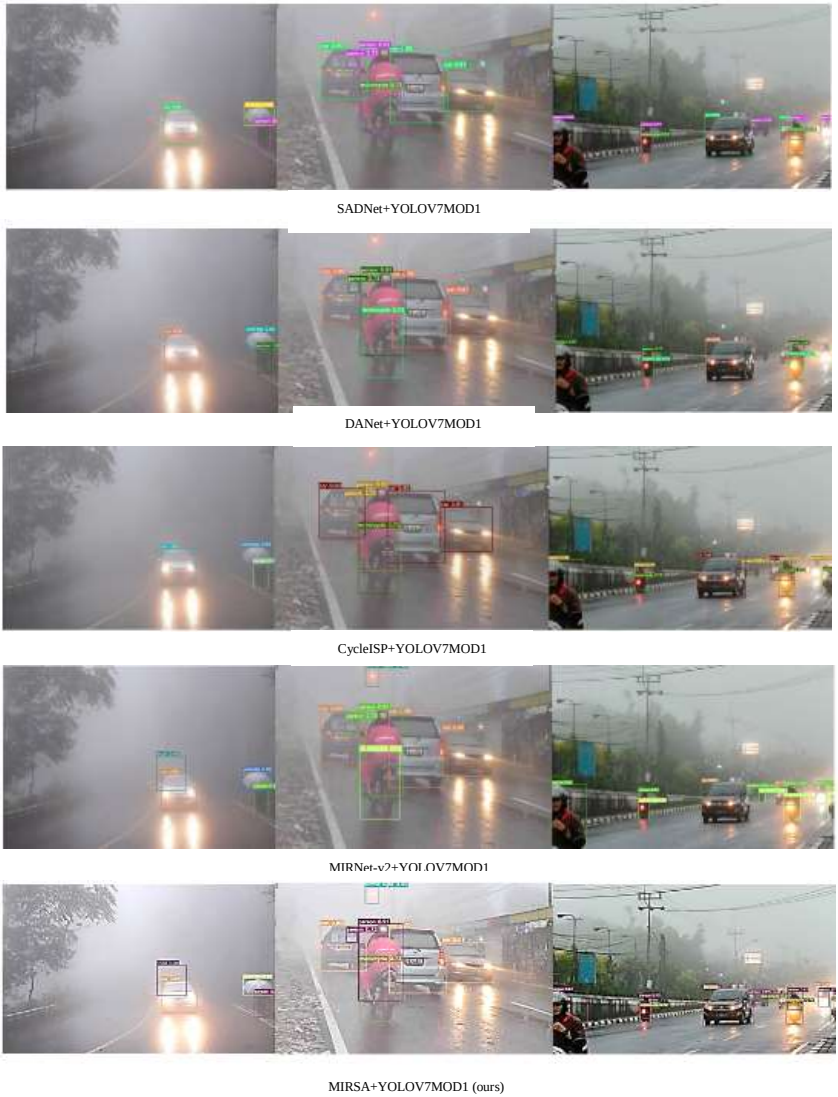
Metode	Berat	Sedang	Ringan
YOLOV7 [34]	0.6487	0.7867	0.8614
YOLOV7MOD [22]	**0.6659**	**0.7923**	**0.8672**

Tabel 7 di atas menyajikan hasil metode Mean Average Precision (mAP) untuk beberapa model deteksi objek, seperti Fast R-CNN, Faster R-CNN, SSD, YOLOV3, YOLOV7, dan YOLOV7MOD. Metode Fast R-CNN menghasilkan mAP sebesar 0.6237, sementara Faster R-CNN dan SSD menunjukkan peningkatan dengan masing-masing nilai 0.6416 dan 0.6434. YOLOV3 memiliki mAP sebesar 0.6148, sedangkan YOLOV7 menunjukkan peningkatan signifikan dengan mAP 0.6487. Pencapaian tertinggi terlihat pada YOLOV7MOD dengan mAP mencapai 0.6659. Analisis terhadap tabel menunjukkan bahwa metode YOLOV7MOD memiliki kinerja deteksi objek yang paling baik, dengan nilai mAP tertinggi. Model ini mengatasi beberapa kendala yang dimiliki oleh model-model sebelumnya, sehingga dapat lebih akurat dalam mendeteksi objek pada berbagai kondisi. Peningkatan performa dari YOLOV7MOD dapat disebabkan oleh pembaruan arsitektur atau strategi pelatihan yang digunakan. Untuk memverifikasi pengaruh modul denoising pada algoritma deteksi objek lainnya, percobaan perbandingan model denoising di atas dengan model deteksi YOLOV7 dan YOLOV7MOD untuk deteksi objek kondisi minim cahaya telah dilakukan. Tabel 8 menunjukkan hasil perhitungan mAP untuk berbagai kombinasi yang disebutkan di atas. Dari tabel tersebut dapat terlihat bahwa akurasi menggunakan metode MIRSA+YOLOV7MOD1 lebih baik jika dibandingkan metode lainnya pada semua kondisi.

Tabel 8 Perbandingan mAP pada beberapa metode deteksi objek gabungan

Metode	Berat	Sedang	Ringan	FPS
SADNet+YOLOV7	0.6536	0.7883	0.8577	30
DANet+YOLOV7	0.6712	0.7911	0.8651	27
CycleISP+YOLOV7	0.6548	0.7816	0.8722	**33**
MIRNet+YOLOV7	0.6679	0.7932	0.8719	32
SADNet+YOLOV7MOD	0.6698	0.7897	0.8593	29
DANet+YOLOV7MOD	0.6763	0.7952	0.8669	26
CycleISP+YOLOV7MOD	0.6781	0.7961	0.8762	30
MIRNet-v2+YOLOV7MOD	0.7136	0.8017	0.8804	28
MIRSA+YOLOV7MOD (ours)	0.7368	0.8235	0.8943	26
MIRSA+YOLOV7MOD1 (ours)	**0.7524**	**0.8391**	**0.9074**	22

Berikut di bawah ini ditampilkan hasil secara kualitatif dari beberapa model yang disebutkan pada Tabel 8.



Gambar 12 Perbandingan kualitatif hasil deteksi objek pada lalu lintas natural pencahayaan rendah yang ditampilkan pada hasil denoising

5. Kesimpulan atau Kontribusi Ilmiah

Berikut beberapa kesimpulan yang dapat diambil dari pengerjaan disertasi ini, yaitu:

1. Pada disertasi ini, diusulkan gabungan metode denoising dan deteksi objek dengan nama MIRSA+YOLOV7MOD+M3CBAM yang merupakan gabungan dari MIRSA dan optimasi YOLOV7MOD dengan CBAM untuk lingkungan lalu lintas beragam dengan pencahayaan rendah dan kondisi cuaca buruk.
2. Metode baru yang diusulkan telah dibandingkan dengan berbagai algoritma denoising dan deteksi objek canggih lainnya. Modul denoising dievaluasi menggunakan metrik evaluasi objektif (PSNR, SSIM) dan metode evaluasi subjektif pada dataset citra lalu lintas pencahayaan rendah yang nyata, sedangkan hasil deteksi objek dievaluasi menggunakan metrik mAP.
3. Hasil eksperimen menunjukkan bahwa metode baru memiliki keunggulan dan efektivitas untuk lingkungan lalu lintas beragam dengan pencahayaan rendah dengan tingkat level noise tinggi, sedang, rendah, memiliki mAP masing-masing sebesar 75.24%, 83.91%, dan 90.74%.
4. Kinerja algoritma deteksi objek MIRSA+YOLOV7MOD dapat ditingkatkan lebih lanjut dengan melakukan estimasi kedalaman layer pada model denoising dan penambahan jumlah dan variasi dataset yang digunakan saat proses pembelajaran.

Kontribusi ilmiah:

- **Pengembangan Metode Persepsi pada Kendaraan Otonom:** Riset ini memberikan kontribusi dalam mengembangkan algoritma yang mampu meningkatkan akurasi dan keandalan deteksi objek dalam kondisi cuaca buruk dan pencahayaan rendah. Hal ini mencakup inovasi dalam algoritma pembelajaran mesin dan *deep learning* yang dapat memproses dan mengintegrasikan data sensor multi-modal untuk memberikan persepsi yang lebih akurat dan *robust* di lingkungan lalu lintas yang beragam.

Tahap selanjutnya dari riset ini adalah mengembangkan algoritma deteksi dan pengenalan objek yang lebih canggih dan *robust*. Ini melibatkan pengumpulan dan anotasi data dalam berbagai kondisi cuaca ekstrem seperti hujan deras, salju, kabut, dan kondisi pencahayaan rendah seperti malam hari atau terowongan. Selain itu, diperlukan integrasi teknologi sensor multi-modal seperti kamera inframerah, lidar, dan radar untuk meningkatkan keandalan sistem dalam mendeteksi dan melacak objek. Algoritma pembelajaran mesin dan *deep learning* akan dilatih menggunakan dataset ini untuk meningkatkan kemampuan persepsi, diikuti dengan pengujian ekstensif dalam simulasi serta uji coba di dunia nyata. Kolaborasi dengan ahli meteorologi dan insinyur pencahayaan akan membantu dalam memahami dan mengatasi tantangan unik yang ditimbulkan oleh kondisi cuaca dan pencahayaan yang bervariasi.

6. Tindak Lanjut

Penelitian ini membuka cakrawala keilmuan baru yang berharga untuk diteliti di masa depan. Salah satunya adalah pengembangan teknologi komunikasi antar kendaraan (V2V) dan antara kendaraan dengan infrastruktur (V2I) untuk meningkatkan koordinasi dan keselamatan di jalan raya dalam kondisi ekstrem. Selain itu, penelitian ini dapat merambah ke bidang ilmu lainnya seperti meteorologi komputasional, di mana data cuaca real-time dan prediksi kondisi jalan dapat diintegrasikan dengan sistem persepsi kendaraan untuk pengambilan keputusan yang lebih baik. Implikasi lainnya termasuk pengembangan sistem bantuan pengemudi yang lebih canggih dan penelitian lebih lanjut dalam pengolahan sinyal dan optimasi algoritma, yang tidak hanya bermanfaat bagi kendaraan otonom tetapi juga bagi aplikasi lain seperti robotika dan sistem pengawasan. Kolaborasi lintas disiplin ini akan mempercepat inovasi teknologi dan meningkatkan keselamatan serta efisiensi transportasi di masa depan.

Riwayat Hidup

Biodata

Nama : Ari Wibowo
Tanggal Lahir : 28 April 1976
Tempat Lahir : Karanganyar-Surakarta
Jenis Kelamin : Laki-laki
Alamat : Jl Ahmad Yani No 1, Batam Center, Batam
Nomer HP : +628127737778
Email : 33218022@mahasiswa.itb.ac.id

Riwayat Pendidikan

- Sarjana Teknik Informatika di Institut Teknologi Bandung, tahun lulus 2000
- Magister Informatika di Institut Teknologi Bandung, tahun lulus 2011
- Research visitor di Politeknik RP Singapore (2017-2018)
- Doktoral Teknik Elektro dan Informatika (2018-sekarang)

Riwayat Pekerjaan

- Asisten D3 Informatika POS Bandung(1998-1999)
- Programmer dan Sistem Analis pada PT. Tritech Consulting Bandung (1999-2000)
- Dosen Politeknik Negeri Batam (2000-sekarang)
- Asisten kuliah Sistem Kendali Cerdas di STEI ITB (2021-2023)
- Asisten Riset Pengembangan Sistem Otonom pada Trem, skema RISPRO LPDP (2021-2024)

Publikasi sebelum doktor

- Prediksi Nasabah Potensial Menggunakan Metode Klasifikasi Pohon Biner (2011-Prosiding Seminar Nasional ISBN 978-979-1194-11-2)

- C2C Marketplace Model in Fishery Product Trading Application Using SMS Gateway (2018-AASEC)

Keahlian
- *Programming, System Analyst, Communication Skill*

Daftar Publikasi Terkait Penelitian

Wibowo, A., Trilaksono, B.R., Hidayat, E.M.I., and Munir, R., 2023, "Object Detection in Dense and Mixed Traffic for Autonomous Vehicles with Modified Yolo". *IEEE Access*. (Published)

Aristo, A.F., Wibowo, A., and Trilaksono, B.R., 2022, "FFCL3D : 3D Object Detection Based on Fuzzy Inference for Camera-LiDAR Fusion", *IEEE Access*. (Submitted 1st Revision)

Wibowo, A., Trilaksono, B.R., Hidayat, E.M.I., and Munir, R., 2024, "Advanced Image Enhancement for Autonomous Vehicles in Low-Light and Adverse Weather Conditions", e-Prime Journal. (Submitted)

Wibowo, A., Trilaksono, B.R., Hidayat, E.M.I., and Munir, R., 2024, "YOLOv3-MOD: Advanced Object Detection and Recognition in Diverse Traffic Conditions for Autonomous Vehicles", ICAE Conference. (Accepted)

Daftar HaKI

1. Paten : Pengembangan Sistem Otonomi Penuh pada Trem
2. Desain Industri : Penempatan Sensor dan Bracket pada Trem Otonom

Ucapan Terima Kasih

Ucapan terima kasih ditujukan kepada:

1. Kementerian Pendidikan dan Kebudayaan, selaku pemberi beasiswa, serta Politeknik Negeri Batam, selaku instansi tempat kerja yang senantiasa memberikan dukungan selama menempuh pendidikan S3.
2. Prof. Dr. Ir. Bambang Riyanto Trilaksono, M.Sc., Egi Muhammad Idris Hidayat, S.T., M.Sc., Ph.D. dan Dr. Ir. Rinaldi Munir M.T.,selaku promotor dan co-promotor yang telah banyak membantu penulis dalam memberikan ide dan saran.
3. Prof. Dr. Armein Z.R. Langi, Dr. Fajar Astuti Hermawati, dan Dr. Nugraha Priya Utama, selaku penguji dan *reviewer* atas semua saran yang diberikan.
4. Istri promovenda Anik Dwi Hartati serta anak, Hasna, Afif, dan Fiza yang selalu memberikan semangat dan doa dalam menyelesaikan pendidikan S3 di ITB.
5. Orang tua promovenda Bpk. Djaimin Sastro Wijaya dan alm. Ibu Warsiyem dan keluarga besar yang selalu memberikan semangat dalam menempuh S3 di ITB.
6. Rekan-rekan seperjuangan, Aria, Hilda, Simon, Agung, Yacub, Wakhyu, Handoko, Ovi, serta tim autonomous trem, Bintang, Aulia, Rini, Atika, Yudo, Farhan, Ibnu, dan Tito. Tidak lupa juga Ibu Nur, Ibu Tita serta semua pihak yang telah membantu promovenda selama menempuh program doktor di ITB.