

**PERBANDINGAN KLASIFIKASI JENIS CITRA AWAN  
MENGUNAKAN ALGORITMA YOLO V8 DAN  
FASTER R-CNN**

**SKRIPSI**

**GEYLFEDRA MATTHEW PANGGABEAN**

**191402065**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**PERBANDINGAN KLASIFIKASI JENIS CITRA AWAN  
MENGUNAKAN ALGORITMA YOLO V8 DAN  
FASTER R-CNN**

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh izajah Sarjana  
Teknologi Informasi**

**GEYLFEDRA MATTHEW PANGGABEAN**

**191402065**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**PERSETUJUAN**

Judul : PERBANDINGAN KLASIFIKASI JENIS CITRA  
AWAN MENGGUNAKAN ALGORITMA YOLO V8  
DAN FASTER R-CNN

Kategori : SKRIPSI

Nama : Geylfedra Matthew Panggabean

Nomor Induk Mahasiswa : 191402065

Program Studi : S-1 Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi

Medan, 4 Juli 2024

Komisi Pembimbing:

Pembimbing 2

Fahrurrozi Lubis B.IT., M.Sc.IT  
NIP. 198610122018052001

Pembimbing 1

Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc.  
NIP. 198603032010121004

Diketahui/disetujui oleh

Program Studi S1- Teknologi Informasi

Ketua,

Dedy Arisandi S1- M.Kom  
NIP. 197908312009121001

**PERNYATAAN****PERBANDINGAN KLASIFIKASI JENIS CITRA AWAN  
MENGUNAKAN ALGORITMA YOLO V8  
DAN FASTER R-CNN****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 4 Juli 2024



Geylfedra Matthew Panggabean

NIM. 191402065

## PERBANDINGAN KLASIFIKASI JENIS CITRA AWAN MENGGUNAKAN ALGORITMA YOLO V8 DAN FASTER R-CNN

### ABSTRAK

Pengamatan dan analisis terhadap awan memiliki peran yang krusial dalam berbagai aspek kehidupan sehari-hari, termasuk dalam prediksi cuaca. Seiring dengan perkembangan teknologi, klasifikasi citra awan menjadi fokus penelitian untuk mendukung aplikasi-aplikasi tersebut. Namun, belum banyak penelitian yang menggunakan pendekatan seperti Faster R-CNN dan YOLO v8 dalam klasifikasi citra awan. Studi ini bertujuan untuk membandingkan efektivitas klasifikasi citra awan menggunakan algoritma YOLO v8 dan Faster R-CNN. Data yang digunakan berasal dari *Singapore Whole sky Imaging CATEGORIES Database* (SWIMCAT) dan mencakup lima kategori awan: *clear sky*, *pattern cloud*, *thick dark cloud*, *thick cloud*, dan *veil*. Melalui serangkaian percobaan dan pengujian, hasil penelitian menunjukkan bahwa algoritma *YOLO v8*, terutama dengan model *pre-trained yolov8m-cls*, memberikan akurasi yang lebih baik daripada *Faster R-CNN* dalam mengklasifikasikan citra awan, dengan akurasi mencapai 97.43% saat diuji sedangkan *Faster R-CNN* mencapai akurasi sebesar 64.10%. Namun, penelitian juga menyoroti bahwa kegagalan dalam klasifikasi dapat terjadi karena karakteristik awan yang memiliki kemiripan, yang dapat dipengaruhi oleh waktu pengambilan citra. Kesimpulannya, klasifikasi citra awan menggunakan *YOLO v8* menunjukkan kinerja yang lebih baik, tetapi pemahaman terhadap karakteristik awan dan pengelolaan waktu pengambilan citra tetap menjadi faktor kunci dalam keberhasilan aplikasi ini.

Kata Kunci: Citra Awan, Klasifikasi, *Faster R-CNN*, *YOLOv8*

## COMPARISON OF CLOUD IMAGE TYPE CLASSIFICATION USING YOLO V8 AND FASTER R-CNN ALGORITHM

### *ABSTRACT*

*Cloud observation and analysis play a crucial role in many aspects of daily life, including weather prediction. Along with the development of technology, cloud image classification has become a focus of research to support these applications. However, not many studies have used approaches such as Faster R-CNN and YOLO v8 in cloud image classification. This study aims to compare the effectiveness of cloud image classification using YOLO v8 and Faster R-CNN algorithms. The data used comes from the Singapore Whole sky Imaging CATegories Database (SWIMCAT) and includes five cloud categories: clear sky, pattern cloud, thick dark cloud, thick cloud, and veil. Through a series of experiments and tests, the results show that the YOLO v8 algorithm, especially with the pre-trained model yolov8m-cls, provides better accuracy than Faster R-CNN in classifying cloud images, with accuracy reaching 97.43% when tested whereas the model when using Faster R-CNN algorithm reached the accuracy of 64.10%. However, the study also highlighted that failures in classification can occur due to similar cloud characteristics, which can be affected by the time of image capture. In conclusion, cloud image classification using YOLO v8 showed better performance, but understanding cloud characteristics and managing image capture time remain key factors in the success of this application.*

*Keyword: Cloud Image, Classification, Faster R-CNN, YOLOv8*

## KATA PENGANTAR

Dengan memanjatkan puji dan syukur penulis kepada Tuhan Yang Maha Esa karena oleh kemurahan dan kesetiaan-Nya penulis dapat menyelesaikan skripsi ini yang merupakan syarat untuk memperoleh gelar Sarjana Komputer, dari Program Studi S1 Teknologi Informasi Universitas Sumatera Utara. Dalam penulisan skripsi ini, penulis dengan rendah hati ingin mengucapkan terima kasih kepada beberapa pihak atas doa, dukungan, dan dorongan yang telah diberikan kepada penulis. Dalam kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Kedua orang tua penulis, Bapak Darwin Panggabean dan Ibu Morita Tobing atas doa, kasih sayang, serta dukungan kepada penulis dari mulai awal pendidikan hingga penulis dapat menyelesaikan tugas akhir ini.
2. Kakak dan abang, Netanya Panggabean dan Andreas Manullang yang mendukung penulis dalam doa dan kasih sayang dari awal Pendidikan hingga penulis dapat menyelesaikan tugas akhir ini.
3. Kepada tante Mona Tobing dan Dame Junita Tobing yang selalu mendoakan penulis dan mendukung penulis dalam pengerjaan tugas akhir ini.
4. Ibu Dr. Maya Silvi Lidya B.Sc., M.Sc selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc. selaku Dosen Pembimbing I dan Bapak Fahrurrozi Lubis B.IT., M.Sc.IT selaku Dosen Pembimbing II yang telah membimbing peneliti dalam penelitian dan penulisan skripsi ini.
6. Ketua Program Studi Teknologi Informasi Bapak Dedy Arisandi ST., M.Kom. Selaku Ketua Program Studi Teknologi Informasi.
7. Dosen Program Studi Teknologi Informasi yang telah memberikan ilmu saat penulis menjalani masa perkuliahan.
8. Staff dan pegawai Fasilkom-TI Universitas Sumatera Utara yang membantu segala urusan administrasi selama perkuliahan.
9. Teman yang sangat membantu peneliti, Aulia Rahman Sihite, Muhammad Zikri Ihsan, Brilliant Jonathan, Christopher Manurung, Monang Limbong, Samuel

Malau, Anggi Yohanes Pardede, dan Timothy Ginting yang selalu menemani dan mendukung penulis dalam penelitian dan penulisan hingga akhir.

10. Teman penulis yang menemani dalam pengerjaan skripsi dan mendukung peneliti dan yang membuat peneliti semakin semangat dalam mengerjakan skripsi, Jordan Tampubolon, Pieter Glorius, Nico Abdi Saputra, Hizkia Silalahi, dan Jonathan Rimbun.
11. Teman penulis pada Angkatan 2020, terutama Albert Lukas Talupan Pangaribuan yang menemani penulis dalam pengerjaan skripsi ini.
12. HUMAS Fasilkom-TI terutama kepada abangda Rhama Permadi, abangda Ahmad Fadhil, adinda Herzinanda Putra, Firman Ramadhani, Dafa Hibrizi, Muhamad Ridho Sirait, dan Donny Adithya.
13. Teman penulis dalam melewati masa organisasi dalam kampus, BPH HIMATIF periode 2022/2023, Aulia Rahman Sihite, Fildzah Alifia, Annisa Putri, dan Nada Salsabila.
14. Teman-teman Teknologi Informasi USU terkhusus Stambuk 2019
15. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu persatu yang telah membantu penyelesaian skripsi ini

Semoga Tuhan Yesus memberkati dan melindungi kepada semua pihak yang telah memberi penulis bantuan dan dukungan kepada penulis dalam menyelesaikan skripsi ini.

Medan, 4 Juli 2024



GEYLFEDRA MATTHEW PANGGABEAN



## DAFTAR ISI

<b>PERSETUJUAN</b>	<b>ii</b>
<b>PERNYATAAN</b>	<b>iii</b>
<b>ABSTRAK</b>	<b>iv</b>
<b><i>ABSTRACT</i></b>	<b>v</b>
<b>KATA PENGANTAR</b>	<b>vi</b>
<b>DAFTAR ISI</b>	<b>viii</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>DAFTAR GAMBAR</b>	<b>xii</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1.Latar Belakang	1
1.2.Rumusan Masalah	3
1.3.Tujuan Penelitian	3
1.4.Batasan Penelitian	4
1.5.Manfaat Penelitian	4
1.6.Metodologi Penelitian	4
1.6.1    Studi literatur	5
1.6.2    Analisis permasalahan	5
1.6.3    Perancangan	5
1.6.4    Implementasi	5
1.6.5    Pengujian program	5
1.6.6    Penyusunan Laporan	5
1.7.Sistematika Penulisan	5
<b>BAB II LANDASAN TEORI</b>	<b>7</b>
2.1.Langit	7
2.2.Awan	7
2.3.Deep Learning	12
2.4.Convolutional Neural Network (CNN)	13
2.5.Region Convolutional Neural Network (R-CNN)	14

2.6.Fast Region Convolutional Neural Network (Fast R-CNN)	14
2.7.Faster Region Convolutional Neural Network (Faster R-CNN)	15
2.8.Perbedaan Antara CNN, R-CNN, Fast R-CNN dan Faster R-CNN	17
2.9.You Only Look Once	18
2.10.You Only Look Once v8 (YOLO V8)	19
2.11.Perbedaan Antara Faster R-CNN Dengan YOLO v8	21
2.12.Confusion Matrix	22
2.13.Penelitian Terdahulu	23
2.14.Perbedaan penelitian	25
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM</b>	<b>27</b>
3.1.Data Penelitian	27
3.2.Analisis Sistem	28
3.2.1 Image Acquisition	28
3.2.2 Image Pre-Processing	29
3.2.3 Model Cloud Type	30
3.3.Perancangan Antarmuka Sistem	32
3.3.1 Tampilan <i>splash-screen</i>	32
3.3.2 Tampilan Home	32
3.3.3 Tampilan upload dari galeri	33
3.3.4 Tampilan <i>cloud type</i>	34
3.3.5 Tampilan Deskripsi	34
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM</b>	<b>35</b>
4.1.Implementasi Sistem	35
4.2.Implementasi Data	35
4.3.Implementasi Perancangan Antarmuka	36
4.3.1 Tampilan splash-screen	36
4.3.2 Tampilan <i>home</i>	37
4.3.3 Tampilan <i>upload</i> dari galeri	37
4.3.4 Tampilan cloud & sky type	37
4.3.5 Tampilan Deskripsi	38
4.4.Pelatihan Sistem	38
4.4.1 You Only Look Once v8 (YOLO V8)	39
4.4.2 Faster R-CNN	41

4.4.3	Perbandingan Hasil Pelatihan Algoritma YOLO V8 dan Faster R-CNN	
	42	
4.5.	Pengujian Sistem	43
4.5.1	Pengujian YOLO v8	43
4.5.2	Pengujian Faster R-CNN	48
4.5.3	Perbandingan hasil pengujian Algoritma YOLO v8 dan Faster R-CNN	54
<b>BAB V</b>	<b>KESIMPULAN DAN SARAN</b>	<b>57</b>
5.1.	Kesimpulan	57
5.2.	Saran	57
<b>DAFTAR PUSTAKA</b>		<b>58</b>

## DAFTAR TABEL

Tabel 2. 1 Perbedaan <i>YOLO V8</i> dengan <i>Faster R-CNN</i>	21
Tabel 3. 1 Rasio jenis awan pada data penelitian	27
Tabel 3. 2 <i>Hyperparameter tuning</i> pada <i>YOLO v8</i>	31
Tabel 3. 3 <i>Hyperparameter tuning</i> pada <i>Faster R-CNN</i>	31
Tabel 4. 1 Hasil proses pelatihan menggunakan yolo8m-cls.pt	39
Tabel 4. 2 Hasil proses pelatihan menggunakan <i>Faster R-CNN</i> (Resnet50)	41
Tabel 4. 3 Hasil Pengujian Sistem menggunakan algoritma <i>YOLO v8</i>	43
Tabel 4. 4 <i>Confusion matrix YOLO v8</i>	45
Tabel 4. 5 Nilai <i>TP</i> , <i>FP</i> , dan <i>FN YOLO v8</i>	46
Tabel 4. 6 Nilai <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score YOLO v8</i>	48
Tabel 4. 7 Tabel pengujian menggunakan <i>Faster R-CNN</i>	49
Tabel 4. 8 <i>Confusion matrix Faster R-CNN</i>	51
Tabel 4. 9 Tabel <i>TP</i> , <i>FP</i> , dan <i>FN Faster R-CNN</i>	52
Tabel 4. 10 Tabel <i>Precision</i> , <i>Recall</i> dan <i>F1-Score Faster R-CNN</i>	54
Tabel 4. 11 Tabel Perbandingan <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i> antara kedua algoritma	54
Tabel 4. 12 Tabel Perbandingan akurasi antara kedua algoritma	55

## DAFTAR GAMBAR

Gambar 2. 1 Awan putih tebal	9
Gambar 2. 2 Awan selimut	9
Gambar 2. 3 Awan Berpola	10
Gambar 2. 4 Langit cerah	11
Gambar 2. 5 Awan gelap	12
Gambar 2. 6 Perbedaan <i>machine learning</i> dengan <i>deep learning</i>	13
Gambar 2. 7 Arsitektur umum <i>CNN</i> (O'Shea & Nash, 2015)	13
Gambar 2. 8 Arsitektur umum <i>R-CNN</i> (Murthy <i>et al.</i> , 2020)	14
Gambar 2. 9 Arsitektur umum <i>Fast R-CNN</i> (Murthy <i>et al.</i> , 2020)	15
Gambar 2. 10 Arsitektur umum <i>Faster R-CNN</i> (Murthy <i>et al.</i> , 2020)	17
Gambar 2. 11 Arsitektur umum <i>YOLO</i> (Redmon <i>et al.</i> , 2015)	19
Gambar 2. 12 Arsitektur umum <i>YOLO v8</i> (Solawetz & Francesco, 2024)	20
Gambar 3. 1 Arsitektur umum	28
Gambar 3. 2 Perbedaan citra sebelum (kiri) dan sesudah (kanan) resizing	29
Gambar 3. 3 Proses <i>resizing</i> ( <i>upsampling</i> )	30
Gambar 3. 4 Kerangka tampilan <i>splash-screen</i>	32
Gambar 3. 5 Kerangka tampilan home	33
Gambar 3. 6 Kerangka tampilan <i>upload</i> dari galeri	33
Gambar 3. 7 Kerangka tampilan Cloud type	34
Gambar 3. 8 Kerangka tampilan deskripsi	34
Gambar 4. 1 Data citra klasifikasi awan	36
Gambar 4. 2 Tampilan <i>splash-screen</i>	36
Gambar 4. 3 Tampilan <i>home</i>	37
Gambar 4. 4 Tampilan <i>upload gallery</i>	37
Gambar 4. 5 Tampilan <i>cloud &amp; sky type</i>	38
Gambar 4. 6 Tampilan Deskripsi	38
Gambar 4. 7 Grafik pelatihan <i>YOLO V8</i>	40
Gambar 4. 8 Grafik pelatihan dengan <i>Faster R-CNN</i> ( <i>Resnet50</i> )	42
Gambar 4. 9 Perbedaan hasil klasifikasi saat fokus citra secara tidak meluas	56

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Dalam kehidupan sehari-hari, cuaca dan iklim sangat berpengaruh atas segala aktivitas manusia. Cuaca merupakan keadaan udara pada saat tertentu dan wilayah tertentu yang relatif sempit dan jangka waktu yang singkat, prediksi cuaca dapat diamati dari berbagai aspek seperti mengamati kondisi langit dan kondisi awan. Awan merupakan kumpulan tetesan air atau partikel es yang tersuspensi di atmosfer dan menutupi lebih dari 50% permukaan bumi (Huang *et al.*, 2018). Awan terdapat di seluruh atmosfer bumi, dimana awan memiliki peran penting pada kondisi cuaca, hidrologi, iklim, susunan kimia udara dan beberapa aplikasi yang bergantung kepada kondisi awan seperti aplikasi bahaya penerbangan di atmosfer dan penggunaan energi matahari pada *solar panel* (Bonkaney *et al.*, 2017; Romano, 2020; Mahajan & Fataniya, 2020; Yuchechen *et al.*, 2020; Prata, 2020). Oleh karena itu penelitian tentang awan sangat berpengaruh untuk pembuatan aplikasi atau alat yang berkaitan dengan iklim dan cuaca.

Dalam penelitian yang terkait analisis awan, data dapat di peroleh dari darat untuk data yang lebih spesifik pada suatu lokal (Sun *et al.*, 2011). Oleh Karena itu, banyak algoritma yang sedang dicanangkan untuk menganalisa awan dari citra langit dengan menggunakan instrumen citra dari darat (S. Liu *et al.*, 2022). Citra langit ini dapat diolah dan digunakan dalam menyelesaikan banyak studi yang berkaitan dengan penggunaan citra awan (Moughyt *et al.*, 2015), yang lebih efektif dibanding dengan menggunakan kemampuan visual manusia.

Perkembangan teknologi membuka jendela kepada perkembangan alat maupun aplikasi yang dapat menggantikan kemampuan manusia. Klasifikasi citra menjadi salah satu bidang teknologi terkini yang dapat menggantikan kemampuan visual manusia (Javidi, 2002). Klasifikasi citra merupakan suatu proses pengelompokan seluruh piksel pada suatu citra kedalam kelompok sehingga diinterpretasikan sebagai suatu properti yang spesifik (Chang & Chiang, 2002).

Penelitian klasifikasi citra awan juga dapat membantu manusia dalam kehidupan sehari-hari, pemanfaatan klasifikasi citra awan dapat membantu manusia sebagai contoh adalah digunakan dalam deteksi cuaca untuk kedepannya dan klasifikasi citra awan dapat di gunakan akademisi yang bergerak dalam bidang meteorologi dan klimatologi dalam tugas klasifikasi awan. Di sisi lain, penelitian ini juga dapat digunakan pada sektor pertanian yang dimana sektor pertanian sangat bergantung pada cuaca untuk kualitas tani yang mereka ingin capai.

Sebelumnya, terdapat beberapa penelitian yang melakukan klasifikasi terhadap citra awan dengan metode yang berbeda-beda. Penelitian yang dilakukan oleh Wanyi Xie *et al.* pada tahun 2020 merupakan salah satu contoh penelitian yang melakukan klasifikasi pada citra langit dengan menggunakan model *SegCloud* dari *deep convolutional neural network* dengan data yang di dapat dari darat dengan menggunakan *all-sky-view camera*. Dari penelitian ini, tercapai hasil akurasi dengan tingkat mencapai 96.24% dimana *SegCloud* mencapai akurasi 96.68% dengan kondisi langit cerah, 95.26% dengan kondisi berawan, dan 99.44% pada kondisi langit mendung.

Pada tahun 2020 silam, Gujanatti *et al.* melakukan penelitian untuk melakukan klasifikasi terhadap awan dengan menggunakan *K-Means Clustering* dan *Content based Image Retrieval Technique* dimana para peneliti mengklasifikasikan awan menjadi 3 kategori, yaitu *high level clouds*, *middle level clouds*, dan *low level clouds*. Pada akhir penelitian, tercapai hasil dengan tingkat akurasi 92.31% untuk *high level clouds*, 76.92% untuk *middle level clouds* dan 90.39% untuk *low level clouds* dengan kondisi hanya tersimpan satu gambar pada pangkalan data. Jika pada pangkalan data tersimpan 2 gambar maka hasil akurasi yang tercapai adalah 96.15% untuk *high level clouds*, 76.92% untuk *middle level clouds*, dan 96.15% untuk *low level clouds*.

Dilihat dari penelitian terdahulu belum terdapat penelitian untuk klasifikasi citra awan dengan menggunakan *Faster Regional Based Convolutional Neural Network (Faster R CNN)* dan dengan menggunakan metode *You Only Look Once v8 (YOLO v8)*. Percobaan akan dilakukan dengan menggunakan beberapa arsitektur untuk menentukan mana arsitektur yang memiliki tingkat akurasi yang memuaskan. Untuk penelitian ini digunakan dataset yang didapat dari penelitian sebelumnya yang bersumber dari *Singapore Whole sky Imaging CATEGORIES Database (SWIMCAT)*. Pada penelitian ini juga akan mengklasifikasikan awan menjadi 5 kelompok yaitu, *clear sky*, *pattern cloud*,

*thick dark cloud*, *thick cloud*, dan *veil*. Judul yang diajukan pada penelitian ini adalah “PERBANDINGAN KLASIFIKASI CITRA AWAN MENGGUNAKAN ALGORITMA *YOLO v8* DAN *FASTER R-CNN*”.

## 1.2. Rumusan Masalah

Berkembangnya algoritma kecerdasan buatan dalam pemrosesan citra digital, mendorong penggunaan algoritma kecerdasan buatan untuk pemrosesan citra digital semakin kerap digunakan dalam rangka mempermudah manusia dalam melakukan tugas klasifikasi. Dalam pemrosesan citra digital, kemampuan *Faster R-CNN* dan *YOLO v8* sangat diakui dalam implementasi dengan tugas klasifikasi. Banyak penelitian yang menggunakan *Faster R-CNN* dan *YOLO v8* sebagai algoritma dalam penelitian, tetapi pada penelitian yang dilakukan oleh Redmon *et al* pada tahun 2016 *YOLO* memiliki performa yang lebih cepat dalam memproses data dibandingkan dengan *Faster R-CNN* dengan menggunakan model *pre-trained* VGG16, tetapi *Faster R-CNN* memiliki keakuratan yang lebih tinggi daripada *YOLO*. Pada tahun 2023, algoritma *YOLO* dikembangkan menjadi *YOLO v8* dengan akurasi dan kecepatan yang lebih baik.

Dari penelitian tersebut, maka perbandingan algoritma *Faster R-CNN* dan *YOLO v8* dibutuhkan untuk mengetahui perbedaan akurasi algoritma tersebut dalam memproses dan memprediksi citra awan dengan menggunakan data citra awan. Algoritma *YOLO v8* dan *Faster R-CNN* akan dibandingkan dalam menyelesaikan masalah klasifikasi yang dapat digunakan oleh praktisi yang mendalami pada bidang klimatologi dan meteorologi.

## 1.3. Tujuan Penelitian

Tujuan peneliti dalam penelitian ini adalah untuk melakukan perbandingan dalam akurasi antar algoritma *YOLO v8* dan *Faster R-CNN* dalam mengklasifikasi awan dengan 5 kelas yaitu *clear sky*, *pattern*, *thick dark*, *thick white*, dan *veil* sehingga dapat diimplementasikan ke aplikasi android agar dapat membantu manusia dalam mengetahui keadaan awan dengan mengurangi faktor kesalahan manusia dalam mengetahui kondisi awan, dan dengan penggunaan implementasi algoritma *YOLO v8* ataupun *Faster R-CNN* diharapkan dapat membantu dalam mengetahui kondisi awan dan memberikan informasi penting yang berguna untuk kegiatan yang terkait dengan awan.



#### 1.4. Batasan Penelitian

Adapun Batasan masalah yang ditentukan untuk membuat penelitian ini terhindar dari pembahasan yang terlalu luas, maka berikut adalah batasan masalah yang telah ditentukan peneliti pada penelitian ini:

1. Data dibatasi dengan penggunaan data *Singapore Whole Sky Imaging Catagories Database* (SWIMCAT).
2. Penelitian ini dibatasi dengan hanya menggunakan algoritma *faster region-based convolutional neural network* dan algoritma *you only look once v8*.
3. Penelitian ini bertujuan untuk mengklasifikasikan awan dari citra langit yang dapat diambil pada langit pada pagi hingga siang hari dimana awan terlihat jelas.
4. Penelitian ini mengklasifikasikan awan sebagai lima kelas, yaitu *clear sky*, *pattern cloud*, *thick dark cloud*, *thick cloud* dan *veil*.
5. Pengujian dilakukan pada level pengenalan objek dalam satu frame saja, oleh dari itu pengujian tidak memperhitungkan perubahan kondisi awan yang berkelanjutan.

#### 1.5. Manfaat Penelitian

Adapun manfaat pada penelitian ini adalah sebagai berikut:

1. Menguji akurasi antar algoritma *YOLO v8* dan *Faster R-CNN* dalam klasifikasi citra awan pada satu lokal dengan baik agar dapat di implementasikan.
2. Menguji akurasi antar algoritma *YOLO v8* dan *Faster R-CNN* dalam melakukan klasifikasi pada citra awan
3. Penelitian ini dapat membuka peluang dalam pengembangan dan penerapan teknologi baru dalam sistem klasifikasi cuaca di masa depan, sehingga dapat membantu meningkatkan akurasi sistem prediksi cuaca.

#### 1.6. Metodologi Penelitian

Berikut adalah arsitektur umum dan gambaran metodologi penelitian pada perbandingan klasifikasi citra awan menggunakan algoritma *YOLO v8* dan *Faster R-CNN*:

#### 1.6.1 Studi literatur

Studi literatur merupakan tahap awal dalam penelitian ini dan pada tahap ini setiap informasi yang dibutuhkan untuk penelitian ini akan dikumpulkan yang dapat berupa jurnal, informasi dari ahli, dan informasi mengenai awan dan algoritma *Faster R-CNN* dan *YOLO v8*.

#### 1.6.2 Analisis permasalahan

Analisis permasalahan merupakan tahap dimana, informasi yang sudah di dapatkan dan dikumpulkan pada tahap sebelumnya dianalisis sesuai dengan keperluan peneliti dalam penelitian ini.

#### 1.6.3 Perancangan

Pada tahap perancangan, arsitektur umum akan dirancang dan pengumpulan data data yang diperlukan pada penelitian ini.

#### 1.6.4 Implementasi

Pada tahap implementasi, algoritma dengan model yang memiliki hasil akurasi yang memuaskan di implementasikan guna mendapatkan hasil dari penelitian yang dibuat.

#### 1.6.5 Pengujian program

Tahap ini dilakukan untuk menguji sistem yang telah dibentuk untuk memastikan sistem yang sudah dibuat dapat digunakan dan bekerja secara baik sehingga dapat mencapai hasil dari penelitian.

#### 1.6.6 Penyusunan Laporan

Penyusunan laporan merupakan tahap akhir, dimana proses penelitian yang telah dilakukan oleh peneliti dapat disusun menjadi sebuah laporan yang disertai dengan dokumentasi sehingga menampilkan hasil akhir pada penelitian.

### 1.7. Sistematika Penulisan

Pada Penelitian ini, sistematika penulisan terdiri atas lima bagian, yaitu:

## **BAB I: PENDAHULUAN**

Pada bab ini peneliti menjelaskan serta menjabarkan hal – hal yang penting dan berkaitan yang mendasari terlaksananya penelitian ini. Bab ini terdiri dari beberapa subbab yaitu, latar belakang dari penelitian, rumusan masalah yang dihadapi dan menjadi dasar untuk penelitian, tujuan penelitian dilaksanakan, batasan penelitian yang ditetapkan dalam penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

## **BAB II: LANDASAN TEORI**

Pada bab ini, peneliti memaparkan teori – teori yang berkaitan yang dapat membantu menjelaskan tentang penelitian yang dilaksanakan. Teori – teori yang dipaparkan berkaitan dengan objek dan algoritma yang digunakan pada penelitian antara lain langit, awan, *deep learning*, *convolutional neural network*, *region convolutional neural network*, *fast region convolutional neural network*, *faster region convolutional neural network*, perbedaan antara *CNN*, *R-CNN*, *Fast R-CNN*, *Faster R-CNN*, *You Only Look Once*, *You Only Look Once v8*, Perbedaan antara *Faster R-CNN* dengan *YOLO v8* dan Penelitian Terdahulu.

## **BAB III: ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini, peneliti menguraikan dan menjelaskan mengenai sistem yang dibangun pada penelitian ini. Hal – hal yang mendukung penelitian ini seperti, data yang digunakan, arsitektur umum, proses preprocessing, pembangunan sistem, dan perancangan antarmuka sistem penelitian.

## **BAB IV: IMPLEMENTASI DAN PENGUJIAN SISTEM**

Pada bab ini, peneliti membahas hasil implementasi dan pengujian dari sistem yang sudah dirancang seperti yang sudah dipaparkan pada bab sebelumnya.

## **BAB V: KESIMPULAN DAN SARAN**

Pada Bab ini, peneliti menuliskan kesimpulan yang didapat dari penelitian yang dilakukan dan memberikan saran yang dapat dijadikan masukan kepada penelitian berikutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1.   *Langit***

Langit adalah ruang kosong di atas permukaan bumi yang tampak seperti kubah biru atau hitam yang membentang di atas kita. Ini adalah bagian dari atmosfer bumi yang mengandung udara, partikel, dan gas lainnya. Langit tampak berbeda tergantung pada waktu, cuaca, dan lokasi geografis. Langit sering terlihat berwarna biru ketika pagi maupun siang hari, karena udara membiaskan cahaya biru dari sinar matahari lebih banyak dibandingkan cahaya merah (Tyndall, 1868). Dari perspektif klimatologi, langit dapat dijelaskan dalam konteks interaksi antara atmosfer dengan fenomena cuaca dan iklim, dan pada klimatologi langit dapat menjelaskan kondisi cuaca dimana langit merujuk pada penampilan fisik dan kondisi atmosfer di atas permukaan bumi pada suatu waktu tertentu. Ini mencakup jenis dan jumlah awan, keberadaan dan intensitas sinar matahari, serta kemungkinan adanya presipitasi (hujan, salju, atau hujan es).

#### **2.2.   *Awan***

Awan adalah salah satu fenomena meteorologi yang paling umum dan penting, mencakup lebih dari 66% permukaan global (Rossow & Schiffer, 1991; Carslaw *et al.*, 2009; Stephens, 2005; Zhao *et al.*, 2019; Wang & Zhao, 2017). Awan terdiri dari tetesan-tetesan air mikroskopis atau kristal es yang berkumpul dalam atmosfer. Partikel-partikel ini bisa sangat kecil dan tidak terlihat dengan mata telanjang. Awan terbentuk ketika udara lembap dan uap air dalam udara mendingin dan terkondensasi menjadi tetesan-tetesan air atau kristal es. Proses ini sering terjadi di sekitar inti debu atau partikel lain di udara, yang bertindak sebagai titik awal untuk pembentukan tetesan.

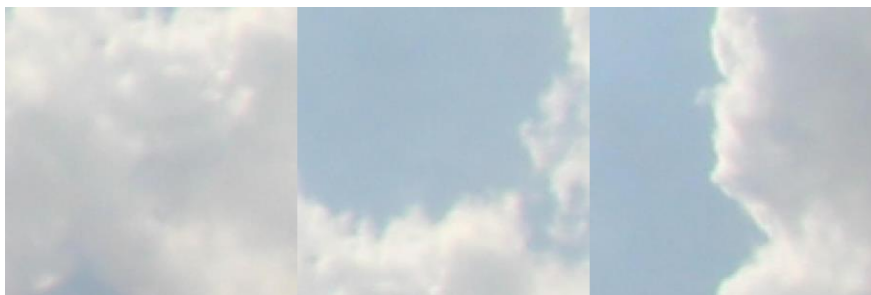
Awan juga memiliki dampak besar pada cuaca karena mereka mempengaruhi jumlah cahaya matahari yang mencapai permukaan bumi, pola angin, dan pembentukan hujan. Awan juga dapat memicu fenomena cuaca ekstrem seperti badai dan tornado. Selama ini, pengamatan awan dilakukan untuk mengamati dan memprediksi cuaca dan mengembangkan model iklim dengan menggunakan alat seperti radar cuaca, satelit dan pengamatan secara manual oleh manusia. Oleh karena itu, penelitian ini sangat berguna dalam memprediksi dan mengamati cuaca menggunakan *image processing* agar tidak mengandalkan pengamatan dari manusia dan dilakukan secara lokal dengan data yang diambil menggunakan kamera yang bersifat diambil dari darat.

Ada beberapa jenis awan yang dapat diidentifikasi berdasarkan penampilan dan ketinggiannya dalam atmosfer. Contoh jenis-jenis awan meliputi :

1. Awan putih tebal (*thick white cloud*)

Awan putih tebal atau yang disebut dengan cumulus adalah salah satu jenis awan yang sering kita lihat di langit. Awan ini memiliki karakteristik tertentu yang membedakannya dari jenis awan lainnya baik dari segi bentuk, ketinggian maupun warna. Awan cumulus memiliki bentuk seperti gundukan putih seperti kapas. Mereka sering terlihat seperti gumpalan awan yang terpisah-pisah di langit. Awan cumulus terbentuk melalui proses konveksi udara. Ini terjadi ketika udara hangat dan lembab naik ke atas karena lebih ringan daripada udara sekitarnya yang lebih dingin. Ketika udara naik, ia mendingin dan uap air dalam udara mulai mengembun, membentuk awan cumulus.

Awan *Cumulus* biasanya terbentuk di ketinggian yang rendah hingga menengah dalam atmosfer, biasanya di bawah 6.000 meter (20.000 kaki) di atas permukaan laut. Awan Cumulus dapat terlihat lebih rendah pada saat pagi atau sore hari dan lebih tinggi selama siang hari ketika matahari menghangatkan atmosfer. Awan *cumulus* seringkali menandakan cuaca yang cerah dan stabil. Awan ini biasanya tidak membawa hujan atau badai. Namun, jika mereka terus berkembang awan cumulus menjadi awan *cumulonimbus* yang lebih besar dan dapat menghasilkan hujan dan badai.

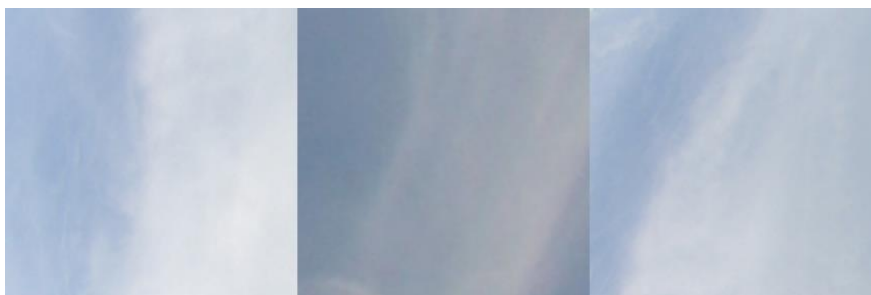


**Gambar 2. 1** Awan putih tebal

## 2. Selimut (*veil*)

Awan yang menyelimuti langit atau yang disebut dengan awan *cirrus* adalah salah satu jenis awan yang memiliki ciri khas tertentu dan biasanya terbentuk di ketinggian yang sangat tinggi di atmosfer. Awan *cirrus* memiliki penampilan yang sangat khas dimana mereka terlihat tipis, bening, dan serat-serat atau garis-garis halus yang menyerupai serat kapas. Awan ini tampak seperti jaring laba-laba di langit dan seringkali membentang dalam bentuk untaian Panjang atau berbentuk bulat-bulat yang terpisah. Awan *cirrus* terbentuk di ketinggian yang sangat tinggi di atmosfer, biasanya di atas 6.000 meter (20.000 kaki) di atas permukaan laut. Mereka seringkali ditemukan di lapisan atmosfer yang disebut troposfer atas atau stratosfer bawah.

Awan *cirrus* terdiri dari kristal-kristal es atau butiran es yang membeku. Mereka terbentuk ketika uap air di atmosfer membeku pada suhu yang sangat rendah, sehingga menciptakan kristal-kristal es yang halus. Awan *cirrus* seringkali terkait dengan cuaca yang cerah dan stabil. Mereka muncul ketika udara di atas sangat kering dan dingin, sehingga jarang membawa hujan. Namun, kehadiran awan *cirrus* juga bisa menjadi petunjuk bahwa perubahan cuaca mungkin akan datang dalam beberapa hari, terutama jika awan *cirrus* berubah menjadi lebih tebal dan berbentuk awan *cirrostratus* atau *cirro-cumulus* yang lebih rendah.

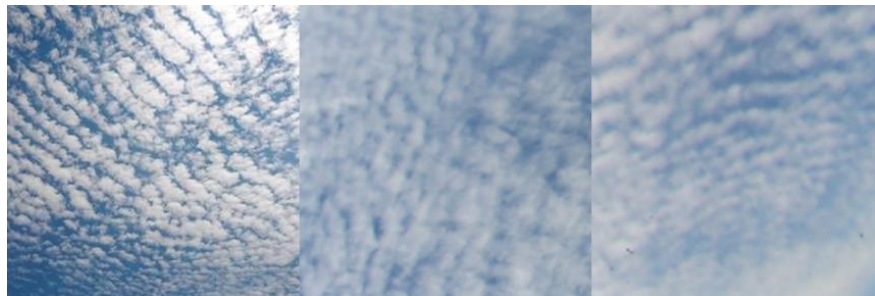


**Gambar 2. 2** Awan selimut

### 3. Berpola (*pattern*)

Awan yang berpola atau yang disebut awan *cirrocumulus* adalah salah satu jenis awan yang terbentuk di lapisan atmosfer yang sangat tinggi. Awan *cirrocumulus* memiliki karakteristik tertentu yang membedakannya dari jenis awan lainnya. Awan *cirrocumulus* memiliki penampilan seperti bongkahan awan kecil yang teratur dan padat, mirip dengan butiran kapas yang tersebar di langit. Awan *cirrocumulus* terbentuk di lapisan atmosfer yang sangat tinggi, biasanya di atas 6.000 meter (20.000 kaki) di atas permukaan laut, di mana suhu sangat rendah. Awan *cirrocumulus* berada di lapisan atmosfer yang sama dengan awan *cirrus*, tetapi *cirrocumulus* lebih tebal dan lebih padat.

Awan *cirrocumulus* terdiri dari kristal-kristal es atau butiran es yang membeku. Ini berarti bahwa meskipun mereka sering terlihat tipis, mereka terdiri dari partikel es yang lebih padat daripada tetesan air yang terkandung dalam awan-awan lain. Awan *cirrocumulus* sering kali muncul ketika udara di atas sangat dingin dan kering. Mereka biasanya tidak membawa hujan, dan muncul sebagai tanda cuaca cerah dan stabil. Tetapi, ketika awan *cirrocumulus* menjadi lebih tebal atau muncul dengan awan-awan lain, ini dapat menunjukkan perubahan sistem cuaca yang mendekat.

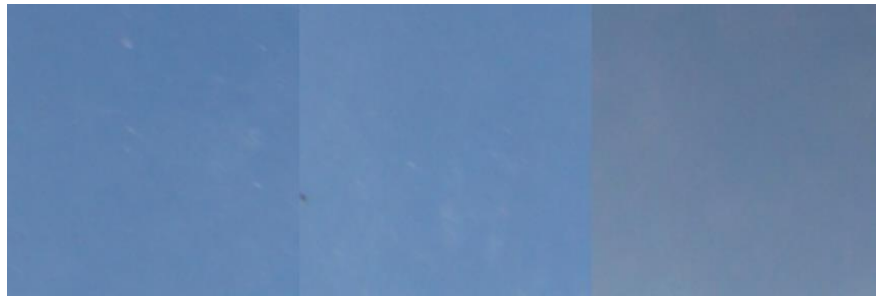


**Gambar 2. 3** Awan Berpola

### 4. Langit cerah (*clear sky*)

Langit cerah adalah kondisi langit di mana tidak ada awan yang menutupi matahari atau bulan, sehingga sinar matahari atau cahaya bulan dapat langsung mencapai permukaan bumi tanpa terhalang oleh awan. Pada siang hari, langit cerah biasanya memiliki warna biru cerah yang intens. Warna ini disebabkan oleh dispersi cahaya matahari oleh molekul-molekul atmosfer yang lebih kecil.

Pada malam hari, langit cerah memungkinkan bulan dan bintang-bintang untuk bersinar dengan jelas.



**Gambar 2. 4** Langit cerah

5. Awan Gelap (*dark cloud*)

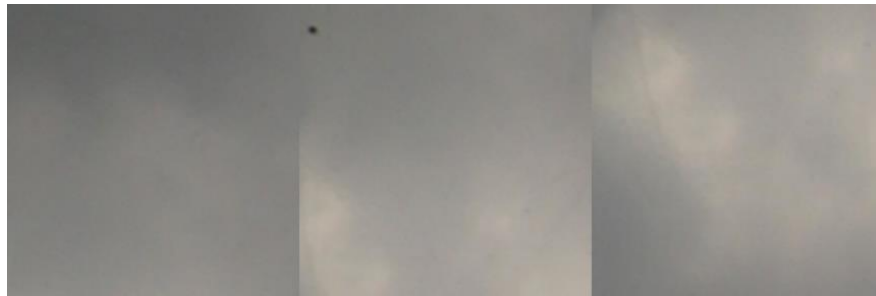
Awan Gelap atau yang disebut dengan awan *cumulonimbus* adalah salah satu jenis awan yang memiliki karakteristik khusus dan seringkali terkait dengan cuaca ekstem, terutama badai petir dan hujan lebat. Awan cumulonimbus memiliki penampilan yang sangat khas. Mereka biasanya terlihat seperti gunung awan yang tinggi dan besar, dengan puncak yang menyerupai kepala dan seringkali diselimuti oleh berkas-berkas awan yang menyerupai kerucut. Mereka juga bisa tampak seperti tumpukan awan dengan tepi yang bergelembung.

Awan *cumulonimbus* adalah salah satu jenis awan tertinggi yang dapat ditemukan di atmosfer. Awan cumulonimbus bisa berketinggian hingga lebih dari 18.000 meter (60.000 kaki) di atas permukaan laut. Ketinggian yang sangat tinggi ini disebabkan oleh naiknya udara panas yang sangat kuat. Awan cumulonimbus terbentuk melalui proses konveksi yang sangat kuat. Ketika udara hangat dan lembap naik dengan cepat ke atas, ia mendingin, dan uap air dalam udara mulai mengembun membentuk awan besar ini. Proses ini dapat dipicu oleh banyak faktor, termasuk pemanasan permukaan atau kondisi topografi tertentu

Awan *cumulonimbus* adalah tanda cuaca yang ekstrem. Awan cumulonimbus seringkali dihubungkan dengan badai petir, hujan lebat, angin kencang, dan tornado. Seiring berkembangnya awan ini, dapat terjadi aktivitas petir dan hujan yang intens. Awan *cumulonimbus* dapat berkembang dengan cepat dan bergerak melintasi langit dengan relatif cepat. Mereka biasanya tidak



bertahan lama, tetapi ketika muncul, mereka dapat menghasilkan cuaca buruk dalam waktu singkat.



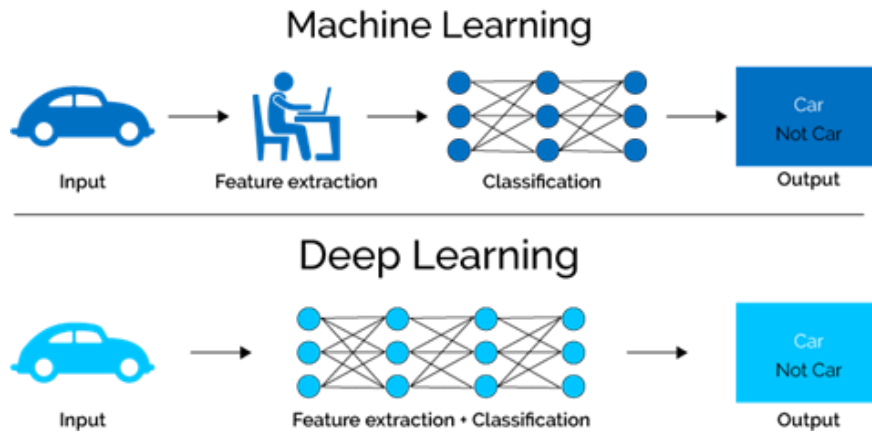
**Gambar 2. 5** Awan gelap

### **2.3. *Deep Learning***

*Deep learning* adalah cabang dari pembelajaran mesin (*machine learning*) yang berfokus pada pengembangan dan penggunaan algoritma yang disebut jaringan saraf tiruan (*artificial neural networks*) untuk mengekstraksi pola dan mengambil keputusan dari data. Istilah "*deep*" dalam *deep learning* merujuk pada kedalaman yang tinggi dari arsitektur jaringan saraf, yang memiliki banyak lapisan (*layer*) di antara *input* dan *output*. *Deep learning* menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah parameter internal yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya (Lv *et al.*, 2022).

*Deep Learning* dan *Machine Learning* adalah dua konsep yang saling terkait dalam dunia kecerdasan buatan, tetapi memiliki perbedaan signifikan dalam hal metode, kompleksitas dan penggunaan.

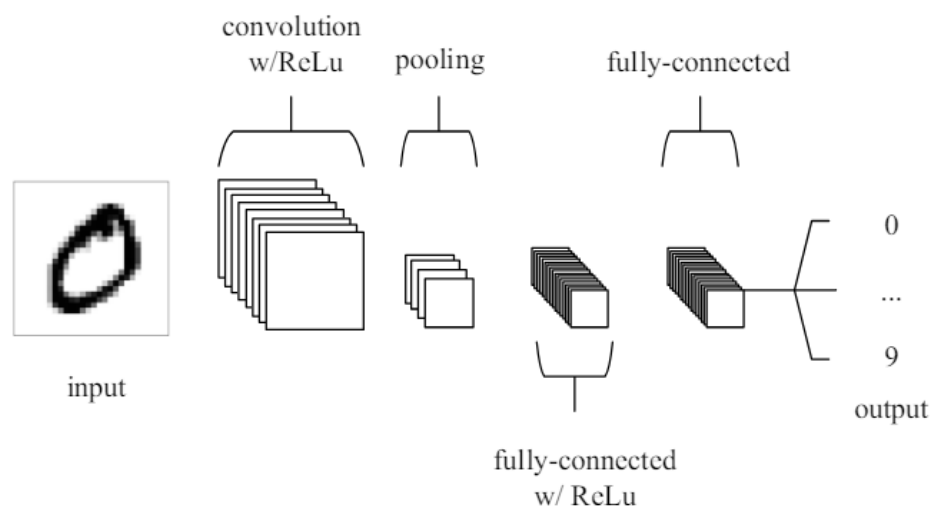
Berikut adalah gambar perbedaan antara *Machine Learning* dan *Deep Learning* agar dapat memudahkan kita dalam memahami perbedaan antar keduanya:



**Gambar 2. 6** Perbedaan *machine learning* dengan *deep learning*  
(Sumber: <https://www.softwaretestinghelp.com>)

#### 2.4. Convolutional Neural Network (CNN)

*Convolutional Neural Network (CNN)* merupakan salah satu jenis dari metode *deep learning* yang mampu untuk melakukan proses pembelajaran mandiri dalam pengenalan objek, ekstraksi objek, dan klasifikasi, serta dapat diterapkan pada citra yang memiliki resolusi tinggi (Zhang *et al.*, 2018). Di mana algoritma ini dirancang secara khusus untuk dapat melakukan proses data *pixel* dan citra visual. *CNN* memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar. Kemudian komputer memperoleh informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan.

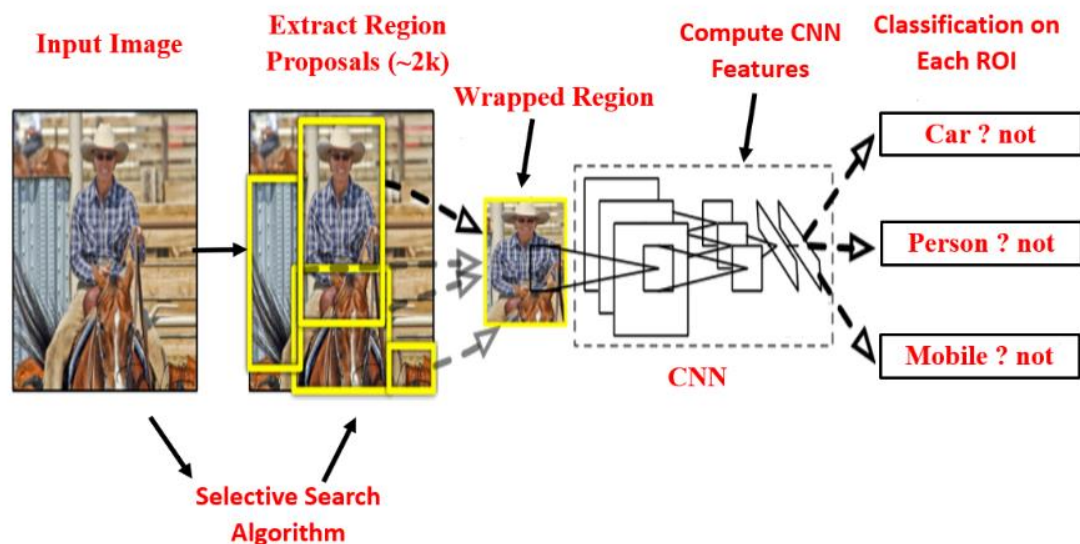


**Gambar 2. 7** Arsitektur umum *CNN* (O'Shea & Nash, 2015)

## 2.5. Region Convolutional Neural Network (R-CNN)

*Region Convolutional Neural Network (R-CNN)* yang termasuk ke dalam salah satu metode *deep learning* merupakan jaringan saraf pertama yang mengajukan wilayah proposal (*region proposals*) untuk mendeteksi objek berdasarkan ekstraksi fitur dan klasifikasi *CNN* yang baik. Di mana objek yang memiliki probabilitas tinggi akan dipilih oleh wilayah proposal untuk menjadi objek dengan menggeser proposal dengan lebar dan tinggi yang berbeda (Du, 2018).

*R-CNN* memiliki tiga modul di dalamnya. Yang pertama adalah untuk menghasilkan wilayah proposal kategori-*independent* dengan menggunakan *selective search*. Yang kedua adalah *convolutional neural network* yang mengekstraksi vektor fitur dari masing-masing wilayah proposal yang telah didapatkan sebelumnya dengan ukuran panjang yang tetap. Yang ketiga adalah *SVM* yang digunakan untuk klasifikasi objek (Girshick *et al.*, 2014).

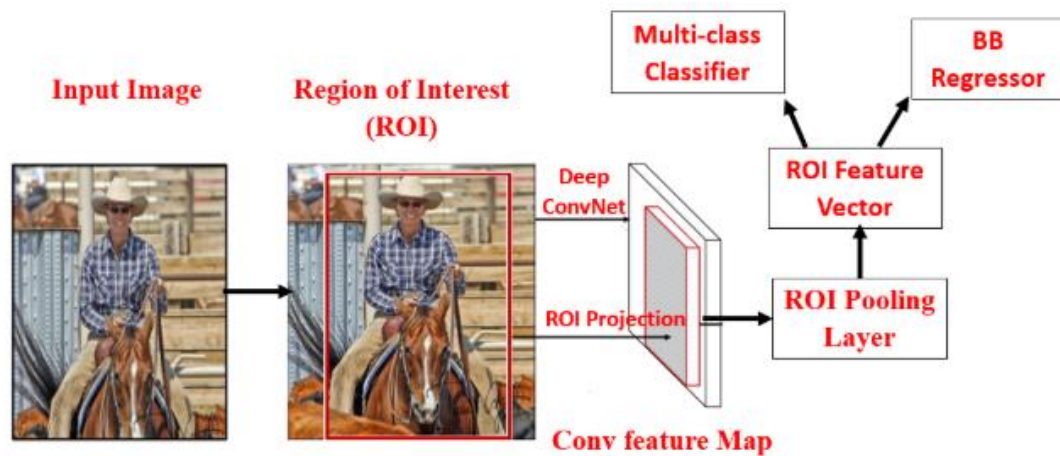


Gambar 2. 8 Arsitektur umum *R-CNN* (Murthy *et al.*, 2020)

## 2.6. Fast Region Convolutional Neural Network (Fast R-CNN)

*Fast R-CNN* merupakan algoritma deteksi objek yang disusulkan pada tahun 2015 oleh Ross Girshick di mana ini merupakan metode lanjutan dari *R-CNN*. *Fast R-CNN* akan mengatasi masalah waktu dalam proses *training* dan *testing*. Proses kerja dari *Fast R-CNN*, yaitu setiap *region* dari *RPN* memiliki *CNN* sebagai fitur ekstraksinya tersendiri. Di mana *Fast R-CNN* hanya memiliki satu *CNN*. Hasil dari *feature map* akan dicocokkan dengan *Region of Interest (ROI)* untuk selanjutnya dilakukan proses

pengklasifikasian kelas. *R-CNN* berfungsi sebagai pengklasifikasi dengan *ROI Pooling* dan lapisan yang terhubung penuh yang digunakan oleh *SVM*. Gambar 2.9 merupakan arsitektur umum yang digunakan pada *fast region-based convolutional neural network*.



**Gambar 2. 9** Arsitektur umum *Fast R-CNN* (Murthy *et al.*, 2020)

## 2.7. *Faster Region Convolutional Neural Network (Faster R-CNN)*

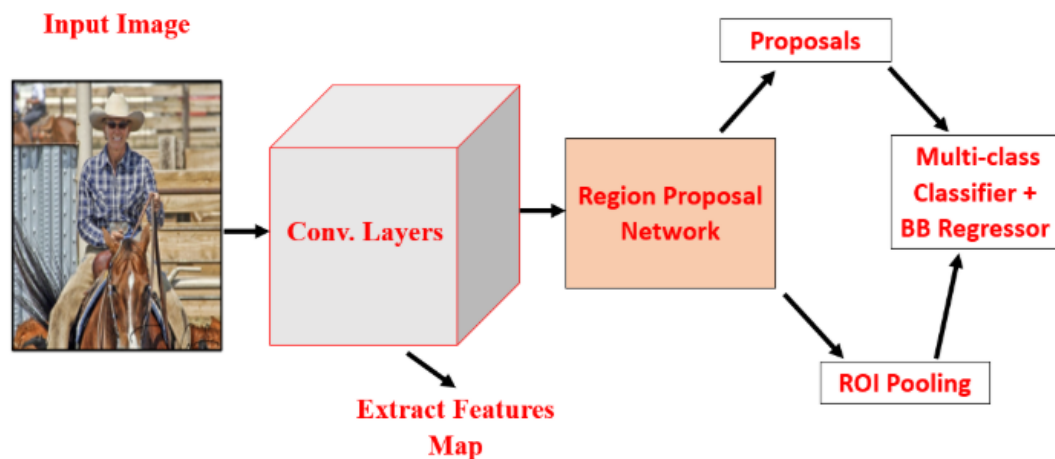
*Faster R-CNN* merupakan pengembangan dari *Fast R-CNN* yang termasuk ke dalam salah satu metode *deep learning* yang digunakan untuk mengenali suatu objek pada citra. Di mana pengenalan dilakukan dengan menelusuri ciri-ciri yang dimiliki oleh objek pada citra. Penelusuran dilakukan melalui sejumlah *layer* melalui proses konvolusi atau *Convolutional Neural Network (CNN)*. *Faster R-CNN* memiliki tiga modul utama, yaitu *deep fully convolutional network* yang mengusulkan *region*, lalu *detector Fast R-CNN* dengan menggunakan *region* yang diusulkan, dan *RPN* yang berfungsi untuk mengarahkan *Fast R-CNN* untuk dapat melihat (Ren *et al.*, 2015).

*Faster R-CNN* memiliki metode *RPN* yang dapat mendeteksi objek dengan kecepatan 0,2 detik per gambar yang dilakukan dengan menelusuri ciri-ciri dari objek pada citra melalui proses konvolusi atau *CNN*. *Faster R-CNN* memiliki tiga fungsi penting dalam pemrosesan, yaitu untuk mengklasifikasikan dan menghasilkan *feature map* dengan *deep neural network*. Lalu untuk menghasilkan proposal wilayah oleh *regional proposal network (RPN)*. Kemudian untuk menemukan regresi dan menambah lapisan *convolutional* oleh *regressor*. Dalam tugas klasifikasi, cara algoritma *Faster R-CNN* dapat dibedakan dengan penggunaan *region proposal network* yang dapat atau tidak digunakan. *RPN* pada tugas klasifikasi dapat disesuaikan dan fitur yang telah

didapat dari proposal wilayah diekstraksi oleh *RoI pooling* untuk mengubah ukuran fitur menjadi tetap. Fitur yang diproses oleh *RoI pooling* diumpankan ke beberapa *fully connected layers* untuk menghasilkan vektor fitur. Hasil dari *fully connected layers* melewati *softmax layer* untuk menghasilkan probabilitas di atas semua kelas yang mungkin. Kelas dengan probabilitas tertinggi diambil sebagai prediksi akhir untuk citra tersebut.

Berikut kode semu untuk proses pelatihan dengan menggunakan algoritma *Faster R-CNN*:

<b>Algoritma I: Pelatihan <i>Faster R-CNN</i></b>		
	<b><i>Model Initialization</i></b>	
	<b><i>INITIALIZE</i></b> <i>Faster R-CNN backbone network model</i> $\mu$	
	<b><i>SET Parameter</i></b> learning rate $\lambda$ , number of classes $\zeta$	
	<b><i>Input:</i></b> Training set $\tau$ , Validation set $v$	
	<b><i>CONFIGURE</i></b> training parameters (batch size $\beta$ , Epoch $\varepsilon$ )	
1	<b><i>FOREACH</i></b> epoch <b><i>FROM</i></b> 1 to $\varepsilon$	
2	<b><i>FOREACH</i></b> $\beta$ in $\tau$	
3	<b><i>FORWARD</i></b> pass to $\mu$	
4	<b><i>CALCULATE</i></b> loss	
5	<b><i>BACKWARD</i></b> pass to model weight $\theta$	
6	<b><i>END FOR</i></b> , $\varepsilon = \varepsilon$	
7	<b><i>VALIDATE</i></b> $\theta$ on $v$	
8	<b><i>OUTPUT</i></b> validation metrics (loss, accuracy, val_loss, val_accuracy)	
9	Simpan <b>model</b> untuk <i>Testing</i>	



**Gambar 2. 10** Arsitektur umum *Faster R-CNN* (Murthy *et al.*, 2020)

## 2.8. Perbedaan Antara CNN, R-CNN, Fast R-CNN dan Faster R-CNN

CNN, R-CNN, Fast R-CNN, dan Faster R-CNN adalah berbagai arsitektur *neural network* yang digunakan dalam tugas deteksi objek, tetapi mereka memiliki perbedaan signifikan dalam cara mereka mengintegrasikan konsep dan mengoptimalkan proses deteksi objek. *CNN* adalah arsitektur dasar untuk klasifikasi gambar, dimana *R-CNN* adalah pendekatan pertama untuk deteksi objek dengan proposal daerah terpisah. *Fast R-CNN* meningkatkan kecepatan dengan memanfaatkan pemrosesan fitur bersama. Dan *Faster R-CNN* adalah evolusi berikutnya dengan mengintegrasikan *Region Proposal Network* (RPN) untuk menghasilkan proposal daerah secara *end-to-end*, membuatnya lebih cepat dan efisien. Dibandingkan dengan *Fast R-CNN*, RPN (*Region Proposal Network*) adalah pengganti metode pencarian selektif yang asli untuk menghasilkan jendela saran oleh *Faster R-CNN*, dan *CNN* dari deteksi objek berbagi dengan *CNN* dari saran jendela (B. Liu *et al.*, 2017).

Adapun perbedaan antara ke empat arsitektur *neural network* adalah sebagai berikut:

### 1. Convolutional Neural Network (CNN)

- a. *CNN* adalah jaringan konvolusi yang dirancang untuk tugas klasifikasi gambar. Mereka mengambil gambar sebagai input dan memetakan gambar tersebut ke label kelas yang sesuai.

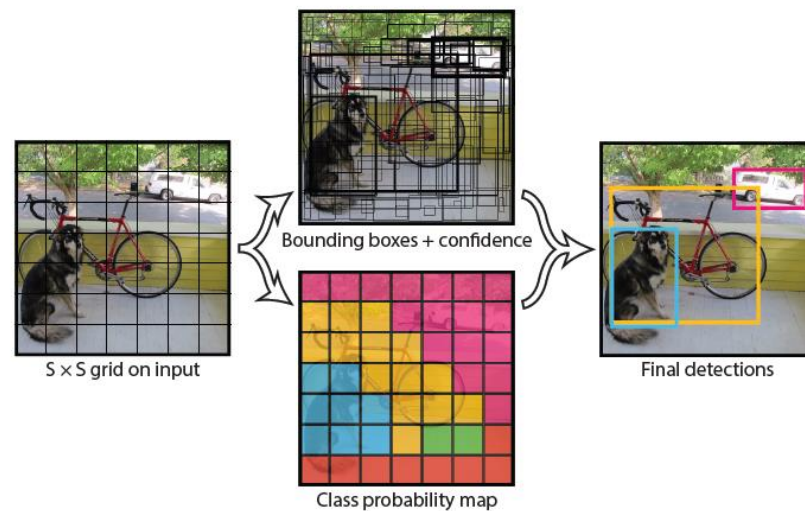
- b. *CNN* tidak memiliki kemampuan untuk mengidentifikaasi atau menentukan lokasi objek dalam gambar
2. *Region-based Convolutional Neural Network (R-CNN)*
  - a. *R-CNN* memecah gambar menjadi sejumlah proposal daerah menggunakan metode seperti *Selective Search*. Kemudian, setiap proposal daerah diproses oleh sebuah *CNN* dan digunakan untuk klasifikasi objek dan regresi koordinat bounding box.
  - b. *R-CNN* lambat dalam pengolahan karena setiap proposal daerah harus diproses secara terpisah oleh *CNN*
3. *Fast Region-based Convolutional Neural Network (Fast R-CNN)*
  - a. *Fast R-CNN* mempercepat proses dengan mengenali bahwa seluruh gambar hanya perlu diproses sekali melalui *CNN* untuk mendapatkan fitur. Kemudian, proposal daerah (RoIs) dihasilkan dan diolah menggunakan *RoI Pooling*. Hal ini mengurangi pengolahan yang redundan.
  - b. *Fast R-CNN* mencoba menggabungkan langkah-langkah pemrosesan menjadi satu arsitektur end-to-end, tetapi masih menggunakan proses terpisah untuk menghasilkan proposal daerah.
4. *Faster Region-based Convolutional Neural Network (Faster R-CNN)*
  - a. Salah satu fitur utama *Faster R-CNN* adalah penggunaan *Region Proposal Network (RPN)* yang terintegrasi dalam jaringan. *RPN* adalah jaringan yang menghasilkan proposal daerah secara otomatis, menggantikan langkah awal *Selective Search* atau metode serupa.
  - b. Dengan integrasi *RPN*, *Faster R-CNN* menjadi lebih cepat dan efisien dalam menghasilkan proposal daerah dan melakukan deteksi objek. Ini adalah langkah signifikan dalam mengurangi waktu komputasi.

## 2.9. *You Only Look Once*

*You Only Look Once* adalah kesatuan model yang diciptakan untuk deteksi objek (Redmon *et al.*, 2015). *You Only Look Once* atau yang disingkat *YOLO* merupakan model objek deteksi yang cepat dan akurat yang membantu komputer dalam mendeteksi objek pada kendaraan, dan dapat membantu manusia melalui gawai dalam memberikan informasi secara langsung (Redmon *et al.*, 2015). Tidak seperti deteksi objek yang



menggunakan *Region Proposal Network*, *YOLO* memproses dengan melihat suatu citra dengan keseluruhan dalam proses *training* dan *testing* yang membuat informasi yang kontekstual terlihat secara implisit (Redmon *et al.*, 2015) dan *YOLO* meminimkan *background error* yang sering terjadi pada metode deteksi objek yang menggunakan *Region Proposal Network*. Adapun cara kerja *YOLO* adalah dengan cara memisahkan gambar ke dalam beberapa *grid* dan memprediksi *bounding box* dan kemungkinan kelas dari gambar yang diproses. Model *YOLO* dapat dilihat pada Gambar 2.11 untuk memudahkan pemahaman dalam cara kerja model *YOLO*.



**Gambar 2. 11** Arsitektur umum *YOLO* (Redmon *et al.*, 2015)

## 2.10. *You Only Look Once v8 (YOLO V8)*

*You Only Look Once v8* atau yang disingkat dengan *YOLO v8* merupakan evolusi dalam metode *YOLO* yang menawarkan pengguna dengan berbagai peningkatan yang komprehensif dan kemampuan serbaguna bagi pengguna (Hussain, 2024). *YOLO v8* dirilis oleh *Ultralytics* pada 10 Januari 2023 dengan peningkatan kecepatan yang bisa memproses sampai dengan 280 FPS dan mempertahankan akurasi yang tinggi.

*YOLO v8* pada tugas klasifikasi memiliki 7 tahap dalam proses klasifikasi citra. Pada tahap awal terdapat pemrosesan input citra setelah tahap pemrosesan input citra, citra yang telah di proses melewati *backbone model* untuk mengekstraksi fitur penting dari citra seperti tepi, tekstur dan pola. Fitur yang telah diekstraksi pada tahap sebelumnya kemudian diproses melalui *pooling layer* guna mereduksi dimensi fitur menjadi satu vektor, lalu vektor fitur diumpankan ke *fully connected layer* untuk menghasilkan prediksi klasifikasi. Setelah diumpankan ke *fully connected layer*, data

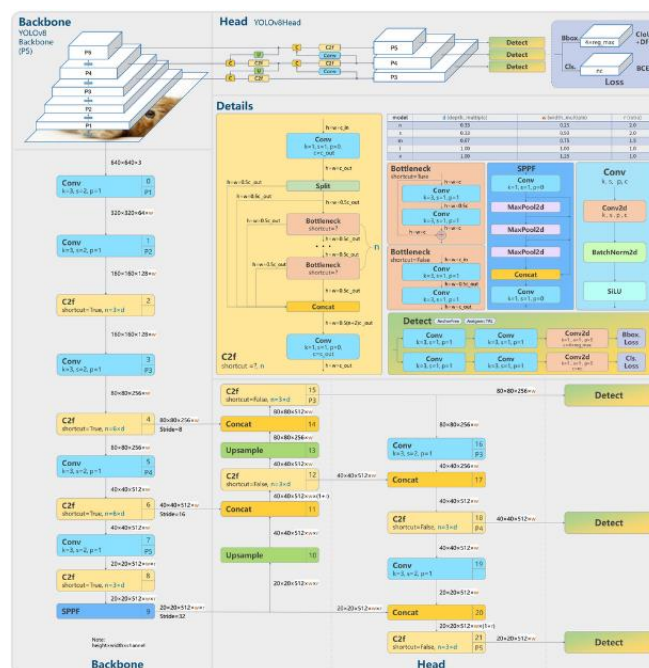


hasil melewati *softmax* layer untuk menghasilkan distribusi probabilitas di atas semua kelas yang mungkin dan kelas dengan probabilitas tertinggi diambil sebagai prediksi akhir untuk gambar tersebut.

Pada gambar 2.12 kita dapat melihat arsitektur dari *YOLO v8* untuk memudahkan pemahaman tentang *YOLO v8*.

Kode semu dalam proses pelatihan menggunakan algoritma *YOLO v8*

Algoritma II: Pelatihan YOLO v8		
	<b>Input:</b> Training set $\tau$ , Validation set $v$ , model $\mu$	
	<b>Configure:</b> learning rate $\lambda$ , batch size $\beta$ , Epoch $\varepsilon$	
1	<b>Foreach</b> epoch <b>FROM</b> 1 to $\varepsilon$	
2		Sampel sebesar $\beta$ dikumpulkan dari $\tau$ , $1 \leq \tau \leq \varepsilon$
3		<b>Forward</b> melalui $\mu$
4		<b>Calculate</b> loss
5		<b>Backward</b> pass untuk update model weights $\vartheta$
6	<b>End For</b> , $\varepsilon = \varepsilon$	
7	<b>Validate</b> $\vartheta$ on $v$	
8	<b>Output</b> validation metrics (top1_acc, top5_acc, train/loss, val/loss)	
9	Simpan <b>model</b> untuk Testing	



**Gambar 2. 12** Arsitektur umum *YOLO* v8 (Solawetz & Francesco, 2024)

### 2.11. Perbedaan Antara Faster R-CNN Dengan YOLO v8

*Faster R-CNN* dan *YOLO v8* memiliki perbedaan yang signifikan walaupun kedua model ini diciptakan untuk *object detection* dan klasifikasi. Pada Tabel 2.1 kita dapat melihat beberapa perbedaan antara *YOLO v8* dengan *Faster R-CNN*.

**Tabel 2. 1** Perbedaan *YOLO V8* dengan *Faster R-CNN*

	<i>Faster R-CNN</i>	<i>YOLO v8</i>
Arsitektur	Memiliki 2 tahapan arsitektur	Memiliki 1 tahap arsitektur
	Menggunakan <i>Region Proposal Network (RPN)</i>	Memprediksi <i>bounding box</i> dan kelas secara langsung
Kecepatan dan Efisiensi	Lebih lambat dikarenakan menggunakan <i>RPN</i>	Sangat cepat dikarenakan pemrosesan dilaksanakan sekali jalan
	Memerlukan komputasi yang lebih banyak	Efisien dalam aplikasi yang <i>real-time</i>
	Lebih cocok dengan pemrosesan data secara offline atau dengan server yang kuat	Dapat digunakan walaupun gawai yang digunakan memiliki kemampuan terbatas
Akurasi	Akurasi tinggi dikarenakan 2 tahapan arsitektur	Akurasi baik namun tidak sebaik <i>Faster R-CNN</i>
	Dapat mencapai <i>Mean Average Precision (mAP)</i> yang tinggi	Dapat mencapai <i>Mean Average Precision (mAP)</i> yang dapat menyeimbangi <i>Faster R-CNN</i> dengan mempertahankan kecepatan
	Cocok digunakan jika tujuan penggunaan condong kepada akurasi	Penggunaan <i>YOLO v8</i> ideal untuk digunakan jika tujuan penggunaan condong kepada <i>real-time</i>

### 2.12. *Confusion Matrix*

*Confusion Matrix* merupakan prediksi dari hasil klasifikasi dengan cara, *output* nilai yang asli dengan *output* nilai yang diinginkan dibanding untuk menghasilkan nilai akurasi. Berikut nilai yang terbentuk dalam *confusion matrix*:

- *True Positive* (TP), yaitu data dengan nilai positif dan dengan prediksi bernilai positif.
- *True Negative* (TN), yaitu data dengan nilai negatif dan dengan prediksi bernilai negatif.
- *False Positive* (FP), yaitu data dengan nilai negatif dan dengan prediksi bernilai positif.
- *False Negative* (FN), yaitu data dengan nilai positif dengan prediksi bernilai negatif.

Ada beberapa metode untuk menghitung *confusion matrix*, yaitu :

#### 1. *Accuracy*

Nilai yang dihasilkan dari jumlah data yang memiliki nilai positif bersama dengan prediksi yang memiliki nilai negatif juga disebut nilai akurasi. Nilai-nilai ini kemudian dibagi dengan jumlah total data yang digunakan dalam penelitian. Akurasi penelitian berkorelasi positif dengan jumlah data yang diprediksi dengan benar.

$$Akurasi = \frac{Jumlah\ data\ yang\ diprediksi\ dengan\ benar}{Jumlah\ keseluruhan\ data} \quad (2.1)$$

#### 2. *Precision*

Nilai peluang yang diprediksi positif dibandingkan dengan nilai keseluruhan hasil yang diprediksi positif dikenal sebagai nilai *Precision*.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

#### 3. *Recall*

Nilai *recall* adalah nilai presentase keberhasilan model dalam mengumpulkan informasi yang diinginkan dan nilai *recall* juga merupakan nilai peluang prediksi benar positif dibandingkan dengan nilai keseluruhan bernilai positif.

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

#### 4. *F1-Score*

*F1-Score* merupakan perbandingan *mean* dari *precision* dan *recall*. Berikut persamaan dalam menghitung *F1-Score*.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.4)$$

### 2.13. *Penelitian Terdahulu*

Gujannati Rudrappa dan Nataraj Vijapur (2020), Melakukan penelitian dengan judul “*Cloud Classification using K-Means Clustering and Content based Image Retrieval Technique*” dimana para peneliti mengklasifikasikan awan menjadi tiga kelas, yaitu: *High Level Cloud*, *Middle Level Cloud*, dan *Low Level Cloud* dengan menggunakan metode *K-Means Clustering* dan *Content based Image Retrieval Technique* (Rudrappa & Vijapur, 2020). Di penelitian ini, *High Level Cloud* mewakili 3 tipe awan yaitu: *Cirrocumulus*, *Cirrus*, dan *Cirrostratus*. Sedangkan *Middle Level Cloud* mewakili 2 tipe awan, yaitu: *altocumulus* dan *altostratus*. Dan yang terakhir, *Low Level Cloud* mewakili 3 tipe awan yaitu: *stratocumulus*, *stratus* dan *nimbostratus*. Dari 6 percobaan yang di lakukan oleh peneliti, terdapat 1 kegagalan klasifikasi pada saat mengklasifikasikan *middle level cloud*. Pada akhir penelitian, tercapai hasil dengan tingkat akurasi 92.31% untuk *high level clouds*, 76.92% untuk *middle level clouds*, dan 90.39% untuk *low level clouds* dengan kondisi hanya tersimpan satu gambar pada pangkalan data. Jika pada pangkalan data tersimpan 2 gambar maka hasil akurasi yang tercapai adalah 96.15% untuk *high level clouds*, 76.92% untuk *middle level clouds*, dan 96.15% untuk *low level clouds*.

Wanyi Xie, Dong Liu, Ming Yang, Shaoqing Chen, Bengu Wang, Zhenzu Wang, Yingwei Xia, Yong Liu, Yiren Wang, dan Chaofan Zhang (2020), melakukan penelitian yang diberi judul “*SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation*” dimana para peneliti membagi klasifikasi menjadi tiga bagian, yaitu: *Clear sky*, *Partial Cloud*, dan *Overcast Sky*. Pada penelitian ini, para peneliti mencoba menggunakan 3

metode yaitu dengan menggunakan metode Otsu, R/B Threshold, dan *SegCloud* (Xie *et al.*, 2020). Data yang digunakan pada penelitian ini adalah WSISEG. Dari 3 metode yang digunakan diatas, didapatkan kesimpulan bahwa akurasi paling tinggi dicapai dengan menggunakan metode *SegCloud* dengan tingkat akurasi mencapai 96.98 % pada *Clear Sky*, 95.26 % pada *Partial Cloud* dan 99.44 % pada *Overcast Sky* dimana *SegCloud* memiliki rata-rata akurasi sebesar 96.24% yang jauh lebih tinggi jika dibanding dengan metode Otsu dan *R/B Threshold*. *SegCloud* merupakan metode yang dikembangkan dari arsitektur VGG-16.

Naufal, M. F., & Kusuma, S. F. (2022). dengan judul penelitian “*Weather Image Classification using Convolutional Neural Network with Transfer Learning*”, Pada penelitian ini, para peneliti menggunakan metode Convolutional Neural Network (CNN) dengan Transfer Learning. Para peneliti menggunakan empat arsitektur CNN, yaitu: *MobileNetV2*, *VGG16*, *DenseNet201*, dan *Xception*. Penelitian ini mengklasifikasikan cuaca menjadi enam kelas, yaitu: berawan, hujan, terang, matahari terbit, bersalju, dan kabut. Dari penelitian ini, terdapat kesimpulan saat menggunakan metode *Xception* memiliki akurasi rata-rata terbaik sebesar 90,21% dan *MobileNetV2* memiliki akurasi rata-rata sebesar 83,51% (Naufal & Kusuma, 2022).

Huang, W., Wang, Y., & Chen, X. (2018). Membahas deteksi awan pada citra jarak jauh beresolusi tinggi dari wilayah perkotaan menggunakan fitur warna dan tepi berdasarkan model dual-warna. Hasil pengujian menunjukkan bahwa metode ini dapat menghasilkan segmentasi awan yang baik dibandingkan beberapa algoritma pembandingnya yaitu SVM dan PRS (Huang *et al.*, 2018).

S. Liu, M. Li, Z. Zhang, X. Cao, dan T.S. Durrani (2020). Penelitian ini berjudul “*Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network*” dan metode yang digunakan dalam penelitian ini adalah Task-based graph convolutional network (TGCN), pada penelitian ini para peneliti menggunakan database yang bernama Ground-based remote sensing cloud database (GRSCD), pada akhir penelitian didapatkan akurasi dari penggunaan metode TGCN dengan akurasi 89.48% (S. Liu *et al.*, 2020).

S. Moughyt, Z. Jaouhari, O. Kadmiri, L. Masmoudi, Z. Kadmiri, dan Y. Zaz (2015) mengusulkan pendekatan yang menggabungkan teknik segmentasi citra dan pengolahan citra untuk estimasi penutupan awan pada citra yang diambil dari kamera dengan metode Otsu's dan multi-objective optimization. Peneliti menggunakan dataset yang diambil dari beberapa lokasi di Maroko dan menunjukkan bahwa pendekatan mereka dapat menghasilkan hasil yang lebih baik dibandingkan dengan teknik segmentasi citra konvensional (Moughyt *et al.*, 2015).

A.Taravat, F. del Frate, C. Cornaro, dan S. Vergari (2015) dengan judul penelitian "*Neural Networks and Support Vector Machine Algorithms for Automatic Cloud Classification of Whole-Sky Ground Based Images*". Pada penelitian ini, para peneliti menggunakan Neural Networks dan SVM untuk mengklasifikasikan citra yang diambil dari kamera. Penelitian ini mengklasifikasikan langit dengan 4 jenis model, yaitu: *clear-sky*, *dark thick cloud*, *high thin cloud* dan *patched cloud*. Pada akhir penelitian, terdapat kesimpulan bahwa performa dari MLP lebih baik dari pada SVM dengan accuracy 95,07% sedangkan SVM sebesar 93,66% (Taravat *et al.*, 2015).

#### 2.14. Perbedaan penelitian

Dari penelitian sebelumnya yang sudah dilakukan, tentunya penelitian ini memiliki perbedaan dan juga persamaan dari penelitian sebelumnya. Perbedaan dari penelitian sebelumnya dapat dilihat dari algoritma yang digunakan. Pada penelitian yang dilakukan oleh Yuzhu Tang *et al.* pada tahun 2020 dengan menggunakan dataset SWIMCAT, para peneliti menggunakan algoritma *Region Covariance Descriptors* dalam klasifikasi citra awan. Pada penelitian tersebut, peneliti mencapai akurasi sebesar lebih dari 98% akurasi dengan menggunakan SWIMCAT dengan *zenithal dataset* yang dikumpulkan dengan *whole-sky infrared cloud-measuring system (WSIRCMS)* (Tang *et al.*, 2020). Algoritma yang akan digunakan dan dibandingkan pada penelitian ini adalah *You Only Look Once v8 (YOLO v8)* dan *Faster Region-based Convolutional Neural Network (Faster R-CNN)* yang belum pernah digunakan pada penelitian-penelitian sebelumnya. Pada penelitian yang dilakukan oleh Xie *et al.* pada tahun 2020 dengan menggunakan *Whole Sky Image SEGmentation (WSISEG)* dengan menggunakan algoritma *SegCloud* yang merupakan metode yang dikembangkan dari VGG16 yang digunakan pada algoritma *convolutional neural*

*network* yang merupakan model yang dikembangkan menjadi *faster r-cnn* yang merupakan salah satu algoritma yang digunakan pada penelitian ini. Penggunaan *YOLO v8* juga merupakan perbedaan penelitian ini dari penelitian sebelumnya dikarenakan algoritma *YOLO v8* yang baru saja dirilis pada tahun 2023. Perbandingan algoritma ini juga bisa membuka kacamata baru tentang kedua algoritma tersebut dan membantu penelitian selanjutnya dalam pengembangan pengolahan citra awan. Pada penelitian ini juga akan menggunakan dataset SWIMCAT (*Singapore Whole Sky Imaging CATegory*) yang tidak digunakan pada algoritma *faster r-cnn* ataupun *YOLO v8* pada penelitian sebelumnya. Dari penelitian ini, diharapkan algoritma yang akan di implementasikan dapat mencapai akurasi diatas 95%.

### BAB III

#### ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini, peneliti akan membahas tentang analisis dan perancangan sistem untuk aplikasi deteksi citra awan dengan menggunakan metode *YOLO V8*. Untuk tahap yang pertama, peneliti akan menjabarkan analisis data yang digunakan dalam penelitian ini, lalu peneliti akan menjabarkan implementasi metode yang digunakan dalam deteksi citra awan. Pada bab ini, peneliti juga akan membahas proses *training* dan *testing*.

##### 3.1. Data Penelitian

Data yang digunakan dalam penelitian ini adalah Singapore Whole sky Imaging CATegories (SWIMCAT) dataset yang berisi 784 gambar dari langit dan awan yang dikategorikan menjadi lima bagian, yaitu *clear sky*, *pattern*, *thick dark clouds*, *thick white clouds*, dan *veil*. Dataset SWIMCAT diambil menggunakan kamera *Wide-Angle High-Resolution Sky Imaging System* yang di kalibrasi untuk mengambil citra langit dari darat (Dev *et al.*, 2015). Lokasi pengambilan dari dataset ini terletak di Singapura. Dari 784 gambar langit tersebut, kategori *clear sky* memiliki 224 gambar, *pattern* memiliki 89 gambar, *thick dark clouds* memiliki 251 gambar, *thick white clouds* memiliki 135 gambar, dan *veil* memiliki 85 gambar.

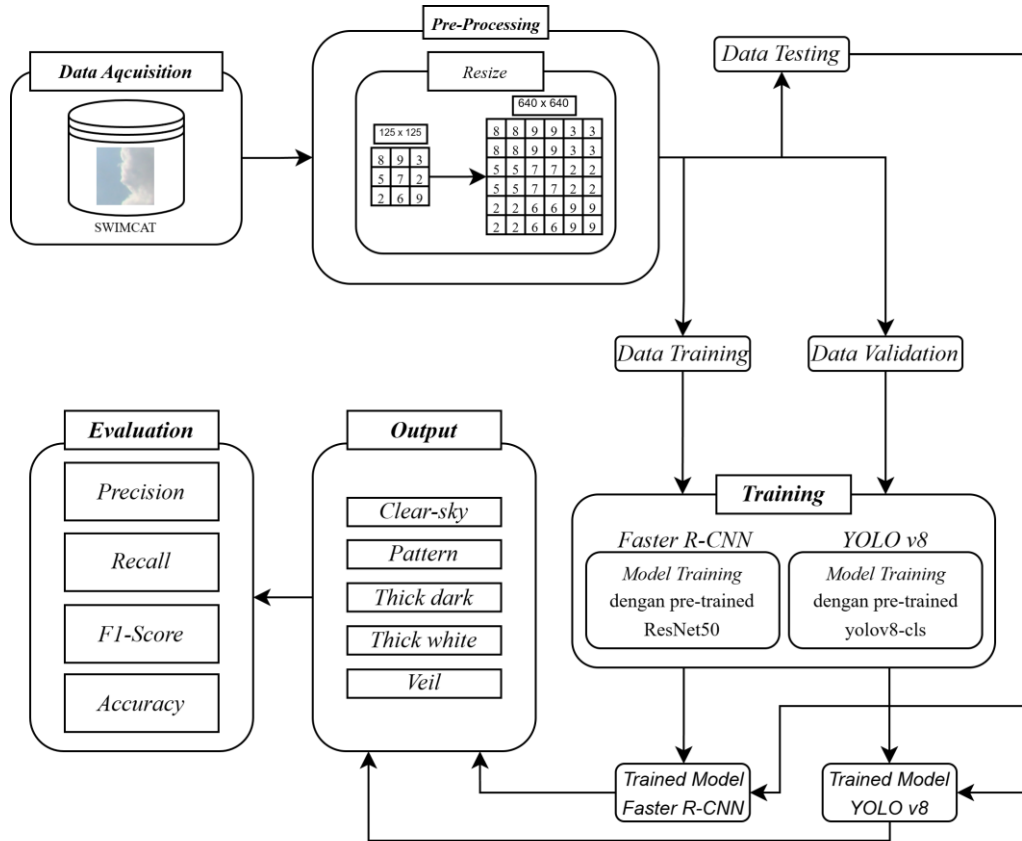
**Tabel 3. 1** Rasio jenis awan pada data penelitian

Jenis Awan	Jumlah Data	Rasio
<i>Clear Sky</i>	224	28.57%
<i>Pattern</i>	89	11.35%
<i>Thick Dark Cloud</i>	251	32.01%
<i>Thick White Cloud</i>	135	17.21%
<i>Veil</i>	85	10.84%



### 3.2. Analisis Sistem

Analisis sistem pada penelitian ini dapat diperhatikan pada arsitektur umum yang tertera pada Gambar 3.1 berikut



**Gambar 3. 1** Arsitektur umum

Pada penelitian ini, data SWIMCAT akan di *pre-process* dan akan di split menjadi *data train*, *data valid* dan *data test*. *Data train* dan *data valid* akan digunakan pada proses *training* tetapi *data test* akan digunakan saat model akan diuji.

#### 3.2.1 Image Acquisition

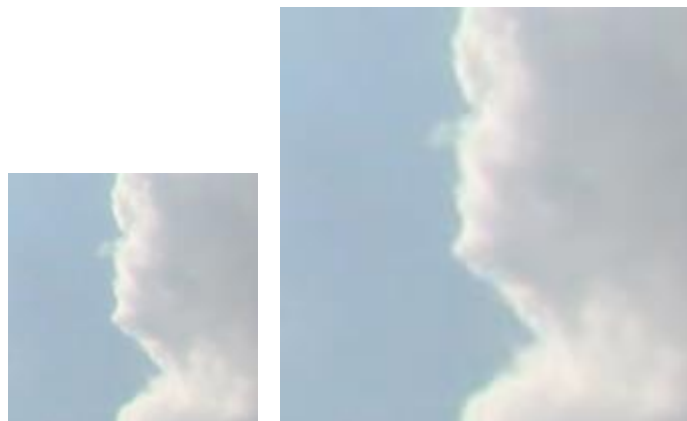
*Image acquisition* pada penelitian ini merupakan tahap dimana data citra langit diambil untuk dijadikan input awal. Data digunakan untuk diimplementasikan pada metode *Faster R-CNN* dan *YOLO v8* untuk mendapatkan perbandingan dalam pengklasifikasian jenis awan.

Pada penelitian ini, peneliti menggunakan dataset *Singapore Whole Sky Imaging CATegories (SWIMCAT)* dataset yang berisi 784 citra awan dan dikategorikan menjadi 5, yaitu *clear sky*, *pattern*, *thick dark cloud*, *thick white cloud*, dan *veil* dimana dataset ini berekstensi PNG. Dataset ini akan di *pre-process* dan akan di pisah menjadi *data train*, *data valid*, dan *data test*

### 3.2.2 Image Pre-Processing

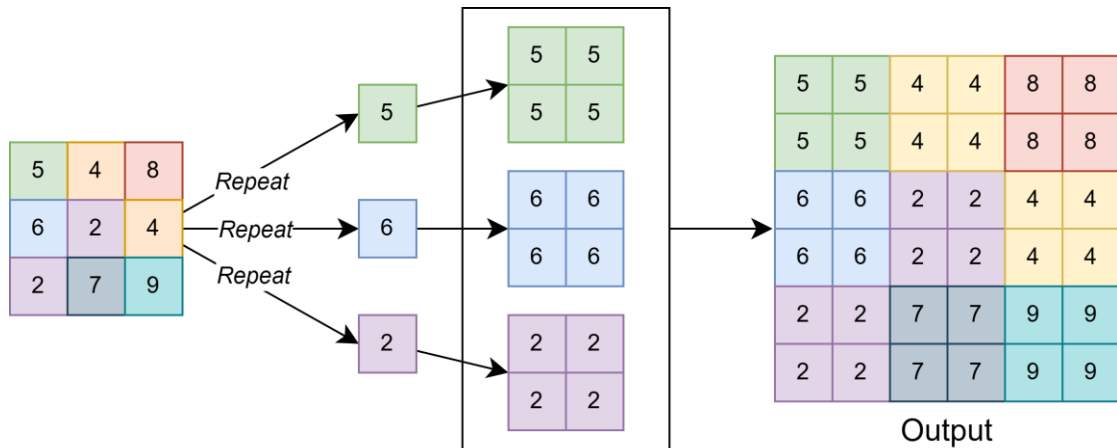
Pada tahap *pre-processing*, dataset SWIMCAT disiapkan untuk merubah kualitas serta bentuk data untuk memudahkan algoritma dalam mempelajari data. *Pre-Processing* yang dilakukan peneliti dalam penelitian ini adalah *resizing* yang dilakukan dengan bantuan *roboflow* dan tidak dilakukan secara manual oleh peneliti.

*Resizing* merupakan metode *pre-processing* yang digunakan pada penelitian ini. Tahap ini dimana ukuran citra diubah sesuai dengan kebutuhan penelitian. Pada penelitian ini, peneliti mengubah ukuran citra menjadi 640 x 640 piksel. Peneliti menetapkan untuk mengubah citra menjadi 640 x 640 piksel dikarenakan peneliti telah mengubah citra menjadi 240 x 240 piksel dan 320 x 320 piksel. Setelah pengujian dan citra dengan 640 x 640 piksel terbukti lebih baik dalam Tingkat akurasi dibandingkan pelatihan model dengan citra dengan ukuran 240 x 240 piksel dan 320 x 320 piksel. Pada penelitian yang dilakukan oleh Talib *et al.*, para peneliti menggunakan citra 640 x 640 untuk mempertahankan keseimbangan diantara efisiensi komputasi dan detail pada citra. Berikut perbedaan sebelum dan sesudah dilakukannya *resizing* pada citra:



**Gambar 3. 2** Perbedaan citra sebelum (kiri) dan sesudah (kanan) *resizing*

*Resizing* dilakukan dengan bantuan *roboflow* sehingga peneliti tidak melakukan *resizing* secara manual. Contoh proses *resizing* pada penelitian:



**Gambar 3. 3** Proses *resizing* (*upsampling*)

Citra dengan 3x3 piksel *diresize* menjadi citra dengan 6x6 dengan mengulang piksel yang berdekatan, sehingga dapat menghasilkan citra yang lebih besar secara ukuran dan dapat memudahkan algoritma untuk mempelajari model.

### 3.2.3 Model Cloud Type

Pada penelitian ini, algoritma yang telah ditentukan adalah *YOLO v8* untuk dibandingkan dengan *Faster R-CNN* dalam klasifikasi citra awan. Setelah proses *pre-processing* dan *data splitting*, *data training* dan *data validation* yang sudah diolah akan diproses dengan algoritma pada saat training. Sebelum dilakukan proses training, ada beberapa proses yang perlu dilakukan.

#### 1. Install Requirement

Pada tahap ini, peneliti memuat *package* yang digunakan untuk proses *training* untuk *YOLO V8* dan *Faster R-CNN* seperti *ultralytics*, *pytorch* dan *dependencies* lain yang digunakan seperti *numpy*, *openCV*, *matplotlib*, *scikit*, *pandas* dan lain lain.

#### 2. Training Model

Klasifikasi pada penelitian ini menggunakan algoritma *YOLO v8* dan dengan menggunakan algoritma *Faster R-CNN*. Pada metode *YOLO V8*, training dibantu dengan menggunakan model *pre-trained yolov8m-cls.pt* dan pada *Faster R-CNN* peneliti menggunakan model *ResNet50*.

Ada proses yang dilakukan sebelum melakukan proses pelatihan model, yaitu *hyperparameter tuning* yang diatur pada algoritma *YOLO v8* maupun *Faster R-CNN*.

*Hyperparameter tuning* dilaksanakan dalam rangka mencari model yang dapat mempelajari data citra dengan baik. Pada algoritma *YOLO v8* *hyperparameter* dicari dengan function *tune* yang disediakan oleh algoritma *YOLO v8*. Pada algoritma *Faster R-CNN* proses pencarian *hyperparameter* dilaksanakan secara manual dengan beberapa percobaan. Proses pelatihan dapat dilihat dengan *pseudocode* pada algoritma I dan algoritma II yang sudah diijelaskan pada bab sebelumnya. Adapun *hyperparameter* yang ditetapkan akan dijelaskan pada tabel 3.2 dan 3.3

**Tabel 3. 2** *Hyperparameter tuning* pada *YOLO v8*

Konfigurasi	Jenis	Parameter
Konfigurasi Model	Optimizer	Adam
	Learning Rate	0.01
Konfigurasi Training	Batch size	8
		16
		32

**Tabel 3. 3** *Hyperparameter tuning* pada *Faster R-CNN*

Konfigurasi	Jenis	Units	parameter
Dense Layer	Activation Layer	16	relu
		5	softmax
Konfigurasi Model	Optimizer		Adam
	Learning Rate		0.01
Konfigurasi Training	Batch size		8
			16
			32

### 3. *Testing Model*

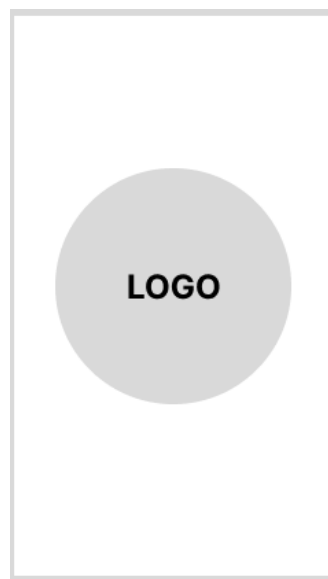
Pada tahap ini, model yang telah di *train* akan di uji dalam klasifikasi jenis citra awan. Model akan melakukan klasifikasi pada data yang telah divalidasi dan pada *data test* sehingga akan didapatkan evaluasi terhadap akurasi model. Setelah proses pengujian atau *testing*, model klasifikasi dengan akurasi terbaik akan di simpan dalam format .pt untuk *YOLO v8* dan .h5 untuk *Faster R-CNN*.

### 3.3. Perancangan Antarmuka Sistem

Pada tahap ini, peneliti melakukan perancangan untuk antarmuka dari aplikasi klasifikasi citra awan yang akan diberi nama SkyIT. Rancangan ini bertujuan sebagai gambaran dari aplikasi yang akan dibangun. Aplikasi SkyIT dirancang oleh peneliti dengan tampilan sebagai berikut, yaitu *splash screen*, halaman *home*, halaman upload dari galeri, dan juga halaman jenis dan informasi mengenai awan.

#### 3.3.1 Tampilan *splash-screen*

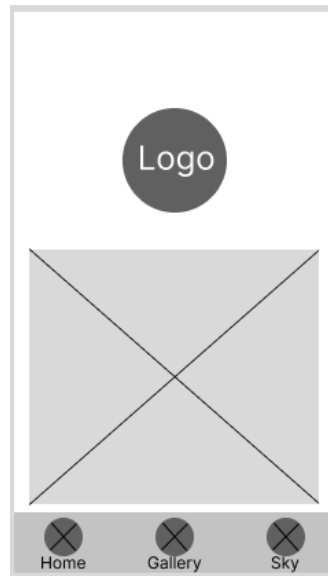
Tampilan *splash-screen* merupakan tampilan yang menyambut pengguna ketika aplikasi dibuka oleh pengguna. Tampilan *splash-screen* yang dirancang oleh peneliti menampilkan logo dari aplikasi SkyIT. Kerangka dari tampilan *splash-screen* dapat dilihat pada gambar 3.4.



**Gambar 3. 4** Kerangka tampilan *splash-screen*

#### 3.3.2 Tampilan *home*

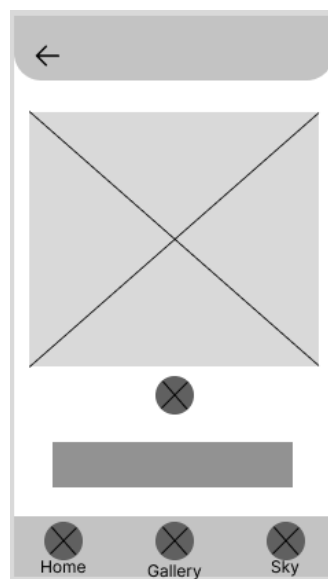
Tampilan home merupakan tampilan utama dari aplikasi SkyIT. Halaman ini dapat dilihat pengguna setelah melewati tampilan *splash-screen*. Pada tampilan ini terdapat bilah navigasi yang terdapat pada bawah layer yang dapat membawa pengguna menuju laman lain seperti laman *gallery* dan jenis-jenis awan. Pada tampilan home juga terdapat petunjuk dalam penggunaan aplikasi sehingga pengguna dapat memahami cara kerja aplikasi ketika pertama membuka aplikasi SkyIT. Kerangka tampilan dari halaman home dapat dilihat pada gambar 3.5.



**Gambar 3. 5** Kerangka tampilan home

### 3.3.3 Tampilan upload dari galeri

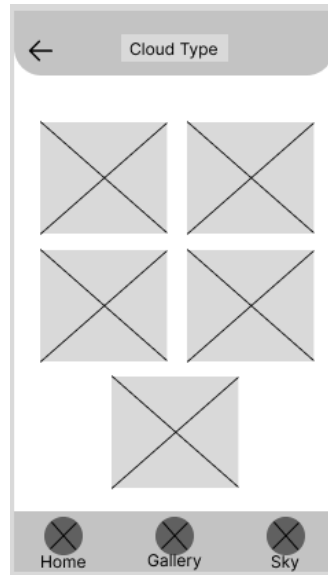
Tampilan ini bertujuan untuk melakukan klasifikasi citra awan dengan cara mengunggah gambar. Pada laman ini terdapat tombol yang dapat mengarahkan pengguna untuk memilih gambar yang terdapat pada gawai pengguna. Setelah pengguna sudah menunggah gambar maka hasil klasifikasi citra awan akan muncul dibawah tombol unggah. Kerangka tampilan laman *upload* dari galeri dapat dilihat pada gambar 3.7.



**Gambar 3. 6** Kerangka tampilan *upload* dari galeri

### 3.3.4 Tampilan *cloud type*

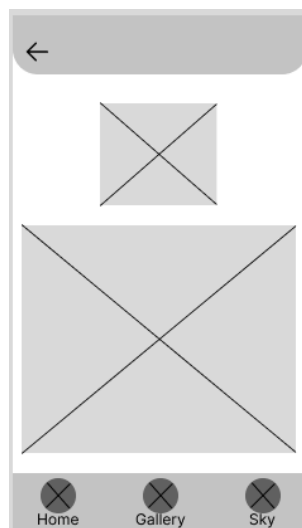
Pada halaman ini, terdapat 5 jenis tipe awan yang dapat diklasifikasikan pada penelitian ini. Laman ini digunakan sebagai jembatan untuk dapat melihat deskripsi tipe-tipe awan yang dapat di klasifikasikan. Kerangka tampilan halaman ini dapat dilihat pada gambar 3.8.



**Gambar 3. 7** Kerangka tampilan Cloud type

### 3.3.5 Tampilan Deskripsi

Pada halaman ini berisikan deskripsi dari langit secara detail. Pada halaman ini terdapat gambar dengan penjelasan dari laman awan yang dituju. Kerangka tampilan laman deskripsi dapat dilihat pada gambar 3.9.



**Gambar 3. 8** Kerangka tampilan deskripsi

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

Pada bab ini, implementasi model terbaik dan perbandingan antara *Faster R-CNN* dan *YOLO v8* untuk klasifikasi citra awan akan dibahas dan di analisis. Bab ini juga berisi rancangan sistem dan proses uji metode pada perangkat android.

#### **4.1. Implementasi Sistem**

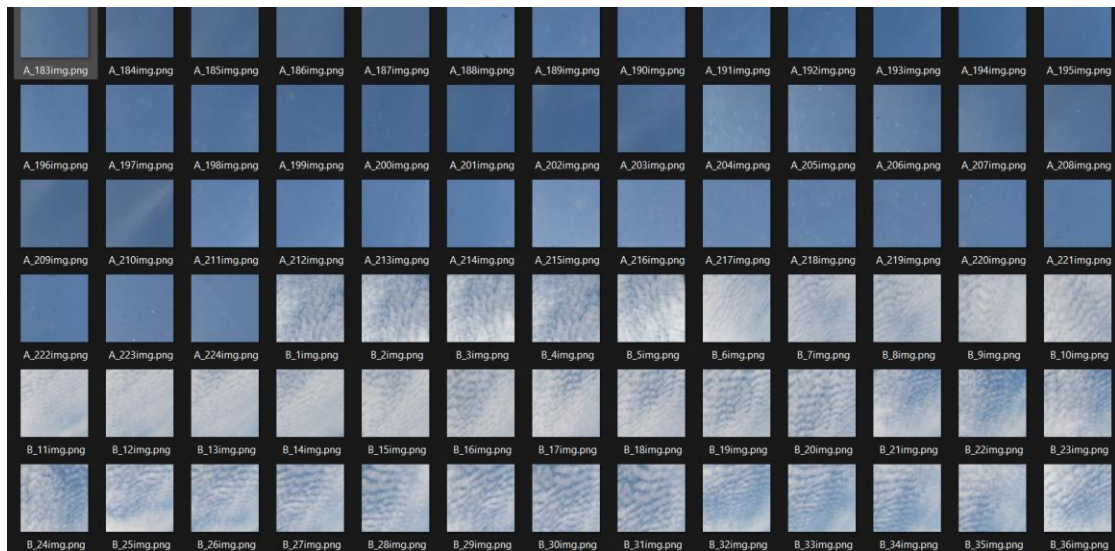
Pada proses ini implementasi metode *YOLO v8* dan *Faster R-CNN* pada penelitian klasifikasi citra awan akan digunakan pada sebuah sistem, berikut hardware dan software yang peneliti gunakan dalam penelitian ini :

1. Laptop Lenovo LAPTOP-T7T2JA6F
2. Processor : AMD Ryzen 5 5600H with Radeon Graphics 3.3GHz
3. GPU : Nvidia GeForce RTX-3050 4 Gb
4. Storage : 512 Gb
5. RAM : 16 Gb
6. OS : Windows 11 Home Single Language 64-bit
7. Google Colab
8. Visual Studio Code

#### **4.2. Implementasi Data**

Pada penelitian ini menggunakan dataset SWIMCAT yang sudah digunakan pada penelitian serupa mengenai citra awan. Data dilabel berdasarkan jenis awan dan dipisah pada folder masing masing. Pada data yang akan di klasifikasikan tetap menggunakan nama klasifikasi yang sudah dibuat oleh pemilik dataset SWIMCAT, yaitu menjadi: *clear sky*, *pattern*, *thick dark*, *thick white*, dan *veil*. Pada gambar 4.1 ditampilkan beberapa contoh data awan.





**Gambar 4. 1** Data citra klasifikasi awan

#### 4.3. Implementasi Perancangan Antarmuka

Pada sub-bab ini peneliti menjelaskan hasil implementasi perancangan antarmuka untuk aplikasi SkyIT yang menggunakan metode terbaik dari hasil perbandingan metode *YOLO v8* dengan *Faster R-CNN* untuk klasifikasi citra awan.

##### 4.3.1 Tampilan splash-screen

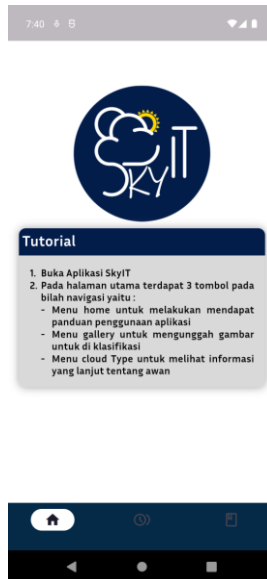
Tampilan splash screen ini dirancang untuk memperkenalkan nama aplikasi. Tampilan ini muncul sebelum pengguna menuju halaman *home*.



**Gambar 4. 2** Tampilan *splash-screen*

#### 4.3.2 Tampilan *home*

Tampilan *home* dirancang untuk menjadi tampilan utama dan pada tampilan *home* memiliki panduan penggunaan aplikasi.



**Gambar 4. 3** Tampilan *home*

#### 4.3.3 Tampilan *upload* dari galeri

Halaman *upload* dari galeri merupakan tampilan yang digunakan dalam proses klasifikasi dengan mengambil citra dari gawai pengguna.



**Gambar 4. 4** Tampilan *upload gallery*

#### 4.3.4 Tampilan cloud & sky type

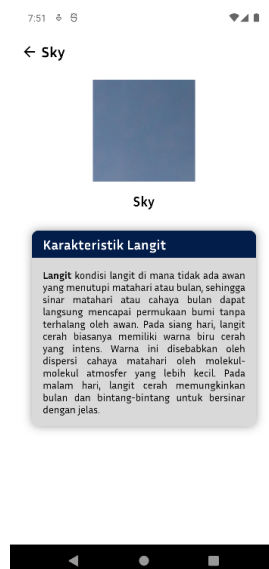
Halaman ini menjadi halaman untuk mengarahkan pengguna menuju halaman deskripsi setiap jenis awan.



**Gambar 4. 5** Tampilan *cloud & sky type*

#### 4.3.5 Tampilan Deskripsi

Halaman ini merupakan halaman yang dapat dituju dari halaman *sky & cloud type* dimana halaman ini memiliki deskripsi dari tipe awan.



**Gambar 4. 6** Tampilan Deskripsi

#### 4.4. Pelatihan Sistem

Pelatihan sistem merupakan langkah untuk melatih model dalam mengenali objek, yang pada penelitian ini merupakan awan agar model dapat mengklasifikasikan awan. Proses ini dibantu dengan penggunaan Google Colab sebagai IDE yang dijalankan dengan

*environment cloud*. Pada sub bab ini pembahasan pelatihan akan dibagi menjadi dua yaitu, pelatihan sistem dengan metode *YOLO v8* dan *Faster R-CNN* dimana pada akhir pembahasan peneliti dapat memilih model mana yang dapat mengenali objek awan lebih akurat dan dapat diimplementasikan kepada aplikasi yang sudah di rancang oleh peneliti. Pada proses pelatihan, metrik yang digunakan sebagai penilaian adalah *train/loss*, *accuracy* dan *val/loss*. *Train/loss* merupakan nilai *loss*, yang merupakan ukuran seberapa jauh prediksi model dari nilai sebenarnya, dihitung pada data pelatihan dan menunjukkan seberapa baik model menyesuaikan diri dengan data pelatihan. Oleh karena itu, *train/loss* memberikan gambaran tentang seberapa cepat dan seberapa baik model belajar dari data pelatihan. *Val/loss* (*validation loss*) merupakan metrik penting yang digunakan untuk mengevaluasi performa model dengan menghitung *loss* menggunakan data *validation* yang membantu mengidentifikasi apakah model bekerja dengan baik pada data yang baru dan tidak hanya pada data pelatihan. *Accuracy* merupakan proporsi prediksi yang benar dibandingkan dengan total jumlah prediksi guna mengukur seberapa sering model klasifikasi membuat prediksi yang benar. Pada proses pelatihan, *batch* dan *epoch* di bedakan dalam mencari model yang terbaik. *Batch* merupakan jumlah sampel yang diproses oleh model dalam satu kali *forward pass* dan *backward pass* selama pelatihan yang dihitung dari jumlah *epochs* yang merupakan satu putaran penuh melalui seluruh dataset pelatihan yang dibagi sesuai dengan ukuran *batch size* yang telah ditentukan.

#### 4.4.1 *You Only Look Once v8 (YOLO V8)*

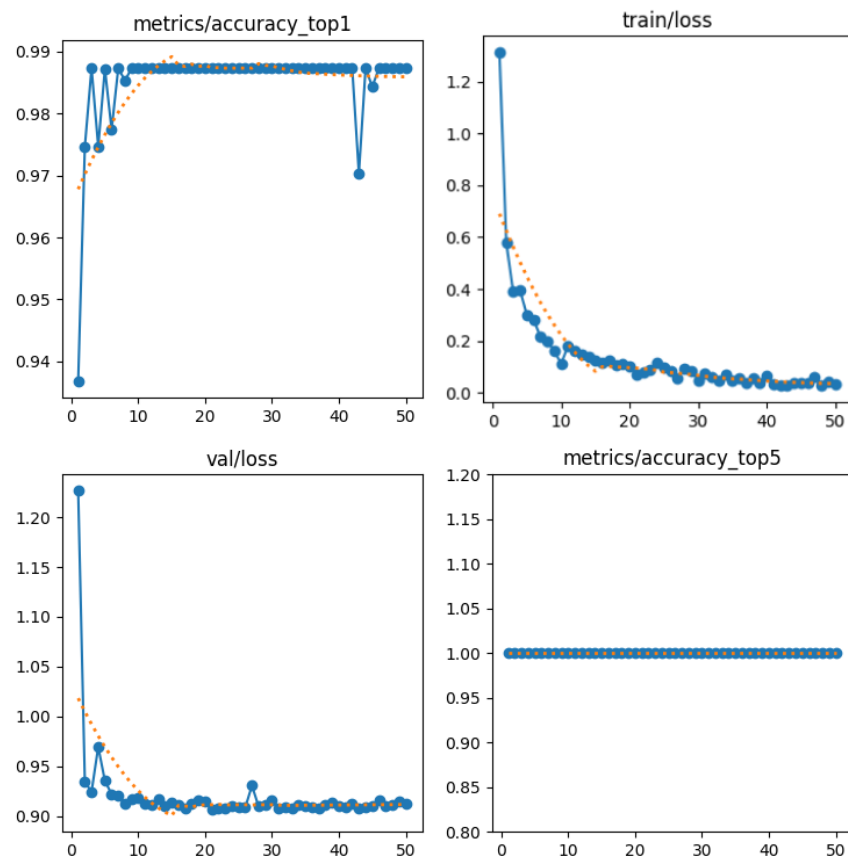
Pada pelatihan sistem dengan algoritma *YOLO v8* peneliti menggunakan *hyperparameter* yang sudah dijabarkan pada tabel 3.2 dan pada proses ini yang berbeda hanya pada jumlah perulangan pelatihan (*epochs*). Peneliti menggunakan model *pre-trained yolov8m-cls.pt* untuk meneliti apakah model *classification* yang didukung dalam algoritma *YOLO v8* dapat mencapai hasil yang memuaskan. Pada tabel 4.1 kita dapat melihat hasil proses pelatihan dengan menggunakan model *pre-trained yolov8m-cls.pt*.

**Tabel 4. 1** Hasil proses pelatihan menggunakan *yolo8m-cls.pt*

<i>Batch</i>	<i>Epoch</i>	<i>Train/loss</i>	<i>Accuracy</i>	<i>Val/loss</i>
8	10	0.23	0.985	0.91
16	10	0.89	0.904	1.07

32	10	0.77	0.955	1.05
8	20	0.72	0.967	1.01
16	20	0.69	0.955	1.02
32	20	0.66	0.975	1.04
8	30	0.56	0.955	1.00
16	30	0.51	0.967	0.98
32	30	0.44	0.967	0.98
8	50	0.34	0.950	0.94
16	50	0.03	0.987	0.91
32	50	0.29	0.975	0.92

Dari proses pelatihan model klasifikasi menggunakan *YOLO v8*, maka didapatkan hasil terbaik dengan akurasi sebesar 0.987, *train/loss* 0.03 dan *val/loss* 0.91 pada pelatihan dengan konfigurasi *epoch* sebanyak 50 dan *batch size* sebanyak 16. Berikut grafik proses pelatihan.



**Gambar 4. 7** Grafik pelatihan *YOLO V8*

Pada gambar 4.7, pada grafik *metrics/accuracy\_top1* terlihat peningkatan akurasi yang cepat menunjukkan bahwa model belajar dengan baik dari data. Sedikit fluktuasi dapat disebabkan oleh variasi dalam kelompok atau proses pelatihan yang tidak stabil. Pada grafik *train/loss* terlihat penurunan tajam dalam *training loss* pada masa-masa awal menunjukkan pembelajaran yang efektif. Penurunan yang lebih lambat dan stabilisasi setelahnya menunjukkan bahwa model ini mulai menyatu. Pada grafik *val/loss* terlihat penurunan *validation loss* menunjukkan bahwa model menggeneralisasi dengan baik pada set validasi. Fluktuasi menunjukkan beberapa variabilitas dalam data validasi. Pada grafik *metrics/accuracy\_top5*, ditunjukkan *accuracy\_top5* yang konsisten dan mengindikasikan bahwa kelas yang benar selalu berada pada 5 prediksi teratas yang merupakan tanda bahwa kinerja model sangat baik untuk metrik ini.

#### 4.4.2 *Faster R-CNN*

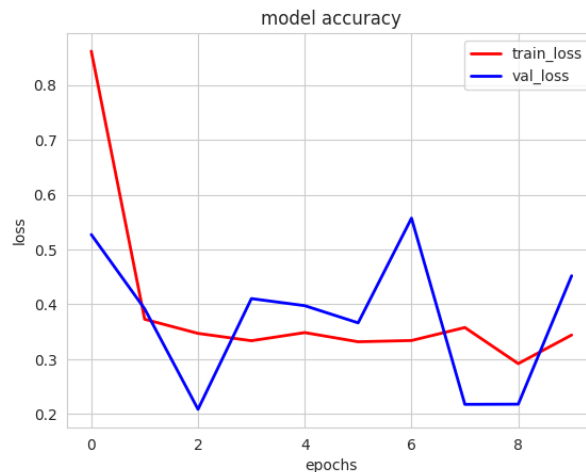
Pada pelatihan sistem dengan algoritma *Faster R-CNN*, peneliti menggunakan konfigurasi yang dijabarkan pada tabel 3.3 dan membedakan *epoch* pada setiap pelatihan. Peneliti menggunakan model *pre-trained Resnet50*. Pada tabel 4.1 kita dapat melihat hasil proses pelatihan dengan menggunakan model *pre-trained ResNet50*.

**Tabel 4. 2** Hasil proses pelatihan menggunakan *Faster R-CNN (ResNet50)*

<i>Batch</i>	<i>Epoch</i>	<i>Train/loss</i>	<i>Accuracy</i>	<i>Val/loss</i>
8	10	0.34	0.912	0.45
16	10	0.21	0.901	0.19
32	10	0.14	0.904	0.20
8	20	0.16	0.899	0.20
16	20	0.15	0.904	0.17
32	20	0.15	0.899	0.16
8	30	0.14	0.904	0.10
16	30	0.15	0.899	0.15
32	30	0.14	0.903	0.18

Dari proses pelatihan model klasifikasi menggunakan *Faster R-CNN* dengan *ResNet50*, maka didapatkan hasil terbaik dengan akurasi sebesar 0.912, train/loss 0.34 dan

val/loss 0.45 pada pelatihan dengan konfigurasi epoch sebanyak 10 dan batch size sebanyak 8. Berikut grafik proses pelatihan.



**Gambar 4. 8** Grafik pelatihan dengan *Faster R-CNN (Resnet50)*

Pada gambar 4.8, grafik pelatihan dengan algoritma *Faster R-CNN* menunjukkan bahwa pada awal pelatihan, baik *train loss* maupun *val loss* sangat tinggi, menunjukkan bahwa model belum belajar dengan baik. Namun, *train\_loss* menurun drastis pada *epoch* pertama, menunjukkan bahwa model dengan cepat mulai belajar pola dari data pelatihan. Setelah itu terjadi penurunan yang signifikan pada *val loss*, menunjukkan peningkatan kinerja model pada data validasi. Ini menjadi indikasi bahwa model berhasil menangkap beberapa fitur penting dari data. *Train loss* mulai stabil dan menurun perlahan, namun *val loss* menunjukkan fluktuasi yang cukup signifikan. Ini menunjukkan bahwa model mengalami beberapa kesulitan dalam generalisasi pada data validasi. Secara keseluruhan, grafik menunjukkan bahwa model mengalami beberapa tantangan dalam hal stabilitas dan generalisasi, yang merupakan hal umum dalam pelatihan model deep learning.

#### 4.4.3 Perbandingan Hasil Pelatihan Algoritma YOLO v8 dan Faster R-CNN

Berdasarkan hasil pelatihan model dengan data yang sama pada *YOLO v8* dan *Faster R-CNN*, didapatkan model terbaik berdasarkan hasil pelatihan dengan menggunakan algoritma *YOLO v8* dengan akurasi sebesar 0.987 dan *val/loss* 0.91. Sedangkan pada pelatihan model menggunakan algoritma *Faster R-CNN* hanya di dapatkan akurasi sebesar 0.912 dan *val/loss* 0.45. maka dengan hasil tersebut peneliti menggunakan model hasil pelatihan dari algoritma *YOLO v8* pada implementasi aplikasi yang dirancang oleh peneliti. Pada hasil pelatihan algoritma tersebut nilai akurasi didapatkan lebih besar pada algoritma


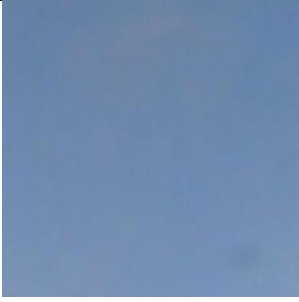

*YOLO v8* tetapi pada nilai *val/loss* algoritma *Faster R-CNN* memiliki nilai yang lebih baik. Pada pelatihan model jika *val/loss* lebih rendah maka dapat disimpulkan bahwa model dapat mengenali objek citra. Oleh karena pernyataan tersebut, peneliti akan membandingkan hasil dari perbandingan akan ditentukan pada pengujian sistem.

#### 4.5. Pengujian Sistem

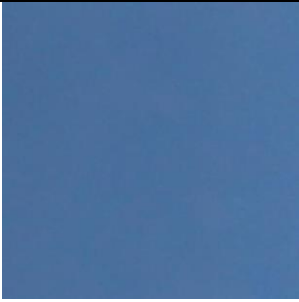


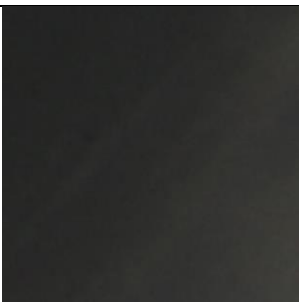
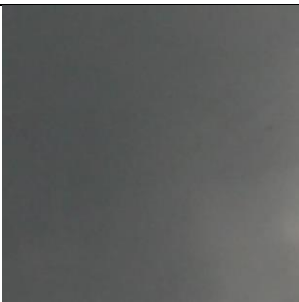
##### 4.5.1 Pengujian *YOLO v8*






Pada tahap pengujian sistem peneliti mengevaluasi proses kemampuan algoritma *YOLO v8*. Uji sistem klasifikasi akan mengevaluasi model untuk mengklasifikasikan kelima jenis awan. Proses evaluasi model menggunakan 10% dari total keseluruhan data yaitu sebanyak 78 data. Hasil pengujian sistem dapat dilihat pada tabel 4.3.

**Tabel 4. 3** Hasil Pengujian Sistem menggunakan algoritma *YOLO v8*

No	Data	Aktual	Prediksi	Keterangan
1		<i>Clear sky</i>	<i>Clear sky</i>	Benar
2		<i>Clear sky</i>	<i>Clear sky</i>	Benar
3		<i>Pattern</i>	<i>Pattern</i>	Benar



4			<i>Clear sky</i>	<i>Clear sky</i>	Benar
5			<i>Veil</i>	<i>Veil</i>	Benar
6			<i>Thick White</i>	<i>Thick White</i>	Benar
7			<i>Thick Dark</i>	<i>Thick Dark</i>	Benar
8			<i>Thick Dark</i>	<i>Thick Dark</i>	Benar

9		<i>Pattern</i>	<i>Pattern</i>	Benar
10		<i>Pattern</i>	<i>Pattern</i>	Benar
11		<i>Clear sky</i>	<i>Clear sky</i>	Benar
12		<i>Thick White</i>	<i>Clear sky</i>	Salah
13		<i>Veil</i>	<i>Clear sky</i>	Salah

Pada tabel 4.3 merupakan sebagian data hasil uji sistem. Setelah melakukan pengujian pada sistem selanjutnya peneliti melakukan evaluasi berdasarkan hasil tersebut menggunakan *confusion matrix*. *Confusion matrix* dapat diperhatikan pada tabel 4.4.

**Tabel 4. 4** *Confusion matrix YOLO v8*

		Aktual					Total
		<i>Sky</i>	<i>Pattern</i>	<i>Thick Dark</i>	<i>Thick White</i>	<i>Veil</i>	
Prediksi	<i>Clear sky</i>	15	0	0	1	1	17
	<i>Pattern</i>	0	15	0	0	0	15
	<i>Thick Dark</i>	0	0	17	0	0	17
	<i>Thick White</i>	0	0	0	16	0	16
	<i>Veil</i>	0	0	0	0	13	13
Total		15	15	17	17	14	78

Dengan memperhatikan *confusion matrix* yang mengevaluasi hasil uji sistem diatas terdapat beberapa kesalahan pada sistem saat di uji. Sistem mengalami kesalahan saat mengklasifikasi tipe *pattern* sebanyak satu kali, dan pada *thick white* sebanyak satu kali. Kesalahan pada sistem tersebut dapat diakibatkan oleh jumlah awan memiliki jumlah yang lebih sedikit dari langit, sehingga sistem mengklasifikasikan *thick white* menjadi *clear sky*. Dan pada *veil* yang diklasifikasikan sebagai *clear sky* diakibatkan oleh karakteristik *veil* yang sangat halus dan memiliki warna yang mirip dengan *clear sky*.

**Tabel 4. 5** Nilai *TP*, *FP*, dan *FN* YOLO v8

Tipe Langit	<i>TP</i>	<i>FP</i>	<i>FN</i>
<i>Clear sky</i>	15	2	0
<i>Pattern</i>	15	0	0
<i>Thick Dark</i>	17	0	0
<i>Thick White</i>	16	0	1
<i>Veil</i>	13	0	1
Total	76	2	2

Berdasarkan tabel 4.5 terdapat 76 data yang bernilai *true positive*, dimana seluruh data tersebut memiliki hasil prediksi yang sesuai dengan data actual. Berdasarkan tabel 4.5 data yang bernilai *false positive* berjumlah 2 data yang memiliki arti data tersebut seharusnya bernilai negatif namun dikenali oleh sistem bernilai positif. Dan terdapat 2 data bernilai *false negative*, dimana data seharusnya bernilai positif namun dikenal bernilai negatif. Seluruh hasil dari *confusion matrix* tersebut akan diolah untuk memperoleh nilai *precision*

menggunakan perhitungan (2.2), *recall* menggunakan perhitungan (2.3), dan *F1-Score* menggunakan perhitungan (2.4).

- *Precision*

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

$$Precision\ Sky = \frac{15}{17} \times 100\% = 88.2\%$$

$$Precision\ Pattern = \frac{15}{15} \times 100\% = 100\%$$

$$Precision\ Thick\ Dark = \frac{17}{17} \times 100\% = 100\%$$

$$Precision\ Thick\ White = \frac{16}{16} \times 100\% = 100\%$$

$$Precision\ Veil = \frac{13}{13} \times 100\% = 100\%$$

- *Recall*

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

$$Recall\ Sky = \frac{15}{15} \times 100\% = 100\%$$

$$Recall\ Pattern = \frac{15}{15} \times 100\% = 100\%$$

$$Recall\ Thick\ Dark = \frac{17}{17} \times 100\% = 100\%$$

$$Recall\ Thick\ White = \frac{16}{17} \times 100\% = 94.1\%$$

$$Recall\ Veil = \frac{13}{14} \times 100\% = 92.8\%$$

- *F1-Score*

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 - Score\ Sky = \left( 2 \times \frac{0.88 \times 1}{0.88 + 1} \right) \times 100\% = 93.61\%$$

$$F1 - Score\ Pattern = \left( 2 \times \frac{1 \times 1}{1 + 1} \right) \times 100\% = 100\%$$

$$F1 - Score Thick Dark = \left( 2 \times \frac{1 \times 1}{1 + 1} \right) \times 100\% = 100\%$$

$$F1 - Score Thick White = \left( 2 \times \frac{1 \times 0.94}{1 + 0.94} \right) \times 100\% = 96.9\%$$

$$F1 - Score Veil = \left( 2 \times \frac{1 \times 0.92}{1 + 0.92} \right) \times 100\% = 95.8\%$$

Pada tabel 4.6 dapat dilihat hasil dari seluruh perhitungan diatas

**Tabel 4. 6** Nilai *Precision*, *Recall*, dan *F1-Score YOLO v8*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Clear sky</i>	88.2%	100%	93.61%
<i>Pattern</i>	100%	100%	100%
<i>Thick Dark</i>	100%	100%	100%
<i>Thick White</i>	100%	94.1%	96.9%
<i>Veil</i>	100%	92.8%	95.8%

Berdasarkan seluruh uji sistem yang digunakan pada model aplikasi klasifikasi citra awan dengan algoritma *YOLO V8* maka didapatkan akurasi sebesar 97.43% berdasarkan perhitungan (2.1)






$$Akurasi = \frac{\text{Jumlah data yang diprediksi dengan benar}}{\text{Jumlah keseluruhan data}}$$


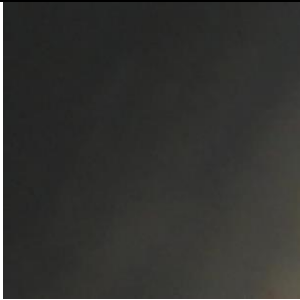

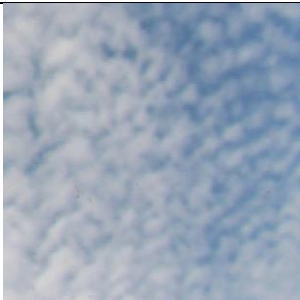

$$Akurasi = \frac{76}{78} \times 100\% = 97.43\%$$




#### 4.5.2 Pengujian *Faster R-CNN*

Pada tahap ini, peneliti menguji model yang didapat setelah melatih algoritma *Faster R-CNN* untuk mengevaluasi proses kemampuan algoritma *Faster R-CNN*. Uji sistem klasifikasi akan mengevaluasi model untuk mengklasifikasikan kelima jenis awan. Proses evaluasi model menggunakan 10% dari total keseluruhan data yaitu sebanyak 78 data sama dengan pengujian yang dilakukan dengan algoritma *YOLO v8*. Hasil pengujian sistem dapat dilihat pada tabel 4.7.

**Tabel 4. 7** Tabel pengujian menggunakan *Faster R-CNN*

No	Data	Aktual	Prediksi	Keterangan
1		<i>Clear sky</i>	<i>Clear sky</i>	Benar
2		<i>Clear sky</i>	<i>Clear sky</i>	Benar
3		<i>Pattern</i>	<i>Pattern</i>	Benar
4		<i>Clear sky</i>	<i>Clear sky</i>	Benar
5		<i>Veil</i>	<i>Thick-white</i>	Salah

6		<i>Thick White</i>	<i>Thick-white</i>	Benar
7		<i>Thick Dark</i>	<i>Thick Dark</i>	Benar
8		<i>Thick Dark</i>	<i>Clear sky</i>	Salah
9		<i>Pattern</i>	<i>Pattern</i>	Benar
10		<i>Pattern</i>	<i>Pattern</i>	Benar

11		<i>Clear sky</i>	<i>Clear sky</i>	Benar
12		<i>Clear sky</i>	<i>Clear sky</i>	Benar
13		<i>Pattern</i>	<i>Thick white</i>	Salah

Pada tabel 4.7 menunjukkan sebagian data hasil uji sistem menggunakan model yang dilatih menggunakan algoritma *Faster R-CNN*. Setelah melakukan pengujian pada sistem selanjutnya peneliti melakukan evaluasi berdasarkan hasil tersebut menggunakan *confusion matrix*. *Confusion matrix* dapat diperhatikan pada tabel 4.8.

**Tabel 4. 8** *Confusion matrix Faster R-CNN*

		Aktual					Total
		<i>Sky</i>	<i>Pattern</i>	<i>Thick Dark</i>	<i>Thick White</i>	<i>Veil</i>	
Prediksi	<i>Clear sky</i>	15	0	10	6	2	33
	<i>Pattern</i>	0	10	0	0	0	10
	<i>Thick Dark</i>	0	0	7	0	0	7
	<i>Thick White</i>	0	5	0	11	1	16
	<i>Veil</i>	0	0	0	4	7	12



<b>Total</b>	15	15	17	21	10	78
--------------	----	----	----	----	----	----

Dengan memperhatikan *confusion matrix* yang mengevaluasi hasil uji sistem diatas terdapat banyak kesalahan pada sistem saat di uji. Sistem mengalami kesalahan saat mengklasifikasi tipe *thick dark* menjadi sky, dan pada *thick white* sebanyak enam kali diprediksi sebagai sky. Pada prediksi *veil* juga mengalami kesalahan pada prediksi menjadi sky dan *thick-white*. Ini dikarenakan oleh karena kemiripan karakteristik dari data dan model tidak dapat mengenali data tersebut. Dengan memperhatikan *confusion matrix* juga kita dapat melihat awan dengan karakteristik yang kuat dapat di prediksi dengan baik seperti *pattern* dan *thick white*.

**Tabel 4. 9** Tabel TP, FP, dan FN *Faster R-CNN*

<b>Tipe Awan</b>	<i>TP</i>	<i>FP</i>	<i>FN</i>
<i>Clear sky</i>	15	18	0
<i>Pattern</i>	10	0	5
<i>Thick Dark</i>	7	0	10
<i>Thick White</i>	11	6	10
<i>Veil</i>	7	4	3
Total	50	28	28

Berdasarkan tabel 4.9 terdapat 50 data yang bernilai *true positive*, dimana seluruh data tersebut memiliki hasil prediksi yang sesuai dengan data aktual. Berdasarkan tabel 4.9 data yang bernilai *false positive* berjumlah 28 data yang memiliki arti data tersebut seharusnya bernilai negatif namun dikenali oleh sistem bernilai positif. Dan terdapat 28 data bernilai *false negative*, dimana data seharusnya bernilai positif namun dikenali bernilai negatif. Seluruh hasil dari *confusion matrix* tersebut akan diolah untuk memperoleh nilai *precision* menggunakan perhitungan (2.2), *recall* menggunakan perhitungan (2.3), dan *F1-Score* menggunakan perhitungan (2.4).

- *Precision*

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

$$Precision\ Sky = \frac{15}{33} \times 100\% = 45.4\%$$

$$\text{Precision Pattern} = \frac{10}{10} \times 100\% = 100\%$$

$$\text{Precision Thick Dark} = \frac{7}{7} \times 100\% = 100\%$$

$$\text{Precision Thick White} = \frac{11}{17} \times 100\% = 64.7\%$$

$$\text{Precision Veil} = \frac{7}{11} \times 100\% = 63.6\%$$

- *Recall*

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

$$\text{Recall Sky} = \frac{10}{15} \times 100\% = 100\%$$

$$\text{Recall Pattern} = \frac{15}{15} \times 100\% = 66.6\%$$

$$\text{Recall Thick Dark} = \frac{7}{17} \times 100\% = 41.1\%$$

$$\text{Recall Thick White} = \frac{11}{21} \times 100\% = 52.3\%$$

$$\text{Recall Veil} = \frac{7}{10} \times 100\% = 70\%$$

- *F1-Score*

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F1 - \text{Score Sky} = \left( 2 \times \frac{0.45 \times 1}{0.45 + 1} \right) \times 100\% = 62.06\%$$

$$F1 - \text{Score Pattern} = \left( 2 \times \frac{1 \times 0.66}{1 + 0.66} \right) \times 100\% = 79.51\%$$

$$F1 - \text{Score Thick Dark} = \left( 2 \times \frac{1 \times 0.41}{1 + 0.41} \right) \times 100\% = 58.15\%$$

$$F1 - \text{Score Thick White} = \left( 2 \times \frac{0.64 \times 1}{0.64 + 1} \right) \times 100\% = 78.04\%$$

$$F1 - \text{Score Veil} = \left( 2 \times \frac{0.63 \times 0.7}{0.63 + 0.7} \right) \times 100\% = 66.31\%$$

Dari perhitungan yang dilakukan diatas, kesimpulan dari *precision*, *recall*, dan *F1-Score* dapat dilihat pada tabel 4.10 berikut.

**Tabel 4. 10** Tabel *Precision*, *Recall* dan *F1-Score* *Faster R-CNN*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Clear sky</i>	45.4%	100%	62.06%
<i>Pattern</i>	100%	66.6%	79.51%
<i>Thick Dark</i>	100%	41.1%	58.15%
<i>Thick White</i>	64.7%	52.3%	78.04%
<i>Veil</i>	63.6%	70%	66.31%

Berdasarkan seluruh uji sistem yang digunakan pada model aplikasi klasifikasi citra awan dengan algoritma *Faster R-CNN* maka didapatkan akurasi sebesar 64.10% berdasarkan perhitungan (2.1)

$$Akurasi = \frac{\text{Jumlah data yang diprediksi dengan benar}}{\text{Jumlah keseluruhan data}}$$

$$Akurasi = \frac{50}{78} \times 100\% = 64.10\%$$

#### 4.5.3 Perbandingan hasil pengujian algoritma *YOLO v8* dan *Faster R-CNN*

Setelah melakukan pengujian terhadap 2 algoritma *YOLO v8* dan *Faster R-CNN* maka pada tabel 4.11 akan memaparkan perbedaan antara kedua algoritma

**Tabel 4. 11** Tabel Perbandingan *Precision*, *Recall*, dan *F1-Score* antara kedua algoritma

<b>Tipe awan</b>	<i>Precision</i>		<i>Recall</i>		<i>F1-Score</i>	
	<i>YOLO v8</i>	<i>Faster R-CNN</i>	<i>YOLO v8</i>	<i>Faster R-CNN</i>	<i>YOLO v8</i>	<i>Faster R-CNN</i>
<i>Clear sky</i>	88.2%	45.4%	100%	100%	93.61%	62.06%
<i>Pattern</i>	100%	100%	100%	66.6%	100%	79.51%
<i>Thick dark</i>	100%	100%	100%	41.1%	100%	58.15%

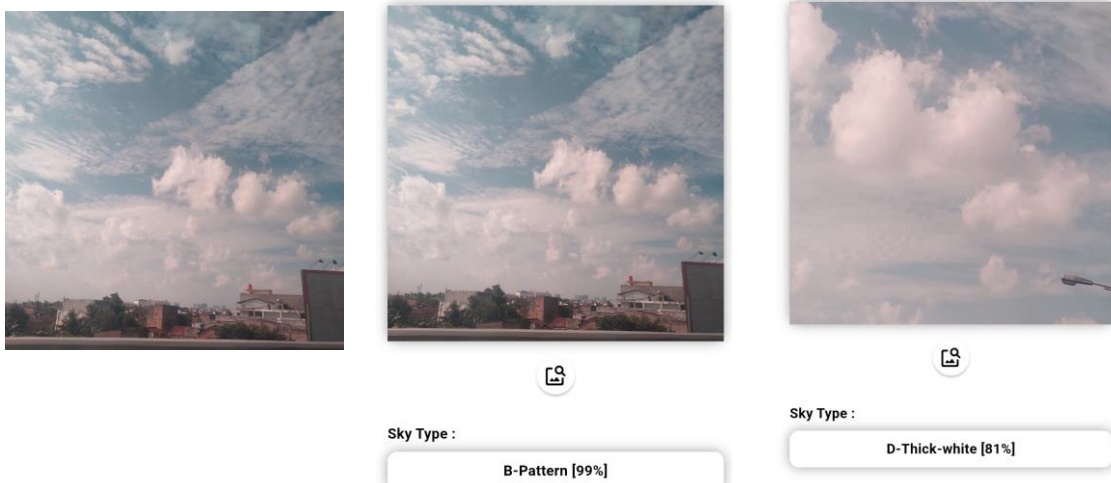
<i>Thick white</i>	100%	64.7%	94.1%	52.3%	96.9%	78.04%
<i>Veil</i>	100%	63.6%	92.8%	70%	95.8%	66.31%

**Tabel 4. 12** Tabel Perbandingan akurasi antara kedua algoritma

	<i>YOLO v8</i>	<i>Faster R-CNN</i>
Akurasi	97.43%	64.10%

Pada tabel 4.11 kita dapat melihat bahwa persentase nilai *precision*, *recall* dan *F1-Score* lebih baik secara keseluruhan dan pada nilai *precision* beberapa tipe awan seperti *pattern* dan *thick dark* didapatkan sama baik dengan *YOLO v8* dan pada persentase nilai *recall* pada tipe awan *clear sky* juga memiliki nilai yang sama baik dengan *YOLO v8* tetapi dikarenakan persentase nilai *recall* lebih besar dibandingkan dengan nilai *precision* maka dapat disimpulkan bahwa model lebih liberal dalam memprediksi positif tetapi menangkap banyak *false positive*. Pada tabel 4.12 kita dapat melihat hasil akurasi dari kedua algoritma dan hasil yang paling baik dalam mengklasifikasi citra awan adalah dengan menggunakan algoritma *YOLO v8* dengan nilai akurasi 97.43% dari model yang di dapat setelah pelatihan menggunakan dataset SWIMCAT. Adapun kegagalan saat klasifikasi dapat disebabkan oleh kualitas citra yang memiliki karakteristik yang mirip sehingga model tidak dapat melakukan prediksi dengan benar yang terjadi saat proses pengujian pada kedua algoritma.

Setelah penerapan model *YOLO v8* pada aplikasi SkyIT yang dirancang oleh peneliti, ada beberapa evaluasi yang peneliti lakukan dalam pengambilan citra untuk di klasifikasi yaitu, citra yang diproses harus memiliki gambar awan secara penuh dan spesifik dalam citra tersebut agar model dapat mengenali citra awan dengan baik.



**Gambar 4. 9** Perbedaan hasil klasifikasi saat fokus citra secara tidak meluas

Pada gambar 4.7 kita dapat melihat perbedaan hasil klasifikasi, dari citra yang sama tetapi pada gambar yang ada di sebelah kanan, citra difokuskan kepada satu jenis awan sehingga hasil klasifikasi yang didapatkan lebih detail dikarenakan citra menangkap lebih dari satu jenis awan dan menemukan awan yang lebih dominan sehingga jika pengguna tidak memproses citra yang memiliki lebih dari satu jenis awan maka hasil klasifikasi model akan menghasilkan hasil klasifikasi awan yang lebih dominan pada citra tersebut.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Adapun kesimpulan yang diperoleh dari hasil penelitian dan pengujian perbandingan algoritma *YOLO v8* dengan *Faster R-CNN* yang telah dilakukan adalah:

1. Hasil akurasi algoritma *YOLO v8* lebih baik daripada akurasi pada algoritma *Faster R-CNN* dalam mengklasifikasikan citra awan dengan mencapai akurasi sebesar 97.43% dimana model yang dicapai oleh model *Faster R-CNN* sebesar 64.10%.
2. Nilai metrik *precision*, *recall*, dan *F1-Score* pada algoritma *YOLO v8* lebih unggul jika dibandingkan dengan nilai metrik pada algoritma *Faster R-CNN*.
3. Algoritma *Faster R-CNN* memiliki belum dapat melakukan klasifikasi pada citra awan dengan menggunakan data SWIMCAT pada pelatihan model dengan baik dibandingkan dengan algoritma *YOLO v8*.
4. Kegagalan dalam klasifikasi dapat disebabkan oleh karakteristik awan yang mirip dan jika adanya lebih dari satu jenis awan pada citra maka model akan memprediksi berdasarkan awan yang lebih dominan pada citra yang diproses.

#### **5.2. Saran**

Adapun saran untuk pengembangan penelitian selanjutnya adalah:

1. Diharapkan penelitian selanjutnya agar membuat perbandingan antara algoritma yang lain agar dapat dibandingkan hasilnya dengan *YOLO v8* dan *Faster R-CNN*.
2. Diharapkan penelitian selanjutnya agar dapat mengimplementasikan model klasifikasi citra awan secara *real time*.

## DAFTAR PUSTAKA

- Bonkaney, A., Madougou, S., & Adamou, R. (2017). Impacts of Cloud Cover and Dust on the Performance of Photovoltaic Module in Niamey. *Journal of Renewable Energy*, 2017, 1–8. <https://doi.org/10.1155/2017/9107502>
- Carslaw, K. S., Harrison, R. G., & Kirkby, J. (2009). Atmospheric science: Cosmic rays, clouds, and climate. In *Science* (Vol. 298, Issue 5599, pp. 1732–1737). <https://doi.org/10.1126/science.1076964>
- Chang, C. I., & Chiang, S. S. (2002). Anomaly detection and classification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 40(6), 1314–1325. <https://doi.org/10.1109/TGRS.2002.800280>
- Dev, S., Lee, Y. H., & Winkler, S. (2015). *Categorization of cloud image patches using an improved texton-based approach*. <http://vintage.winklerbros.net/swimcat.html>.
- Du, J. (2018). Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series*, 1004(1). <https://doi.org/10.1088/1742-6596/1004/1/012029>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. <http://arxiv>.
- Huang, W., Wang, Y., & Chen, X. (2018). Cloud detection for high-resolution remote-sensing images of urban areas using colour and edge features based on dual-colour models. *International Journal of Remote Sensing*, 39(20), 6657–6675. <https://doi.org/10.1080/01431161.2018.1466069>
- Hussain, M. (2024). YOLOv1 to v8: Unveiling Each Variant-A Comprehensive Review of YOLO. *IEEE Access*, 12, 42816–42833. <https://doi.org/10.1109/ACCESS.2024.3378568>
- Javidi, Bahram. (2002). *Image recognition and classification: algorithms, systems, and applications*. Marcel Dekker.
- Liu, B., Zhao, W., & Sun, Q. (2017). *Study Of Object Detection Based On Faster R-CNN*.

- Liu, S., Duan, L., Zhang, Z., Cao, X., & Durrani, T. S. (2022). Ground-Based Remote Sensing Cloud Classification via Context Graph Attention Network. *IEEE Transactions on Geoscience and Remote Sensing*, 60.  
<https://doi.org/10.1109/TGRS.2021.3063255>
- Liu, S., Li, M., Zhang, Z., Cao, X., & Durrani, T. S. (2020). Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network. *Geophysical Research Letters*, 47(5). <https://doi.org/10.1029/2020GL087338>
- Lv, Q., Zhang, S., & Wang, Y. (2022). Deep Learning Model of Image Classification Using Machine Learning. *Advances in Multimedia*, 2022.  
<https://doi.org/10.1155/2022/3351256>
- Mahajan, S., & Fataniya, B. (2020). Cloud detection methodologies: variants and development—a review. *Complex and Intelligent Systems*, 6(2), 251–261.  
<https://doi.org/10.1007/s40747-019-00128-0>
- Moughyt, S., Jaouhari, Z. El, Kadmiri, O. El, Masmoudi, L., Kadmiri, Z. El, & Zaz, Y. (2015). *Cloud coverage estimation using ground based images and segmentation techniques Image Processing: 3D medical image watermarking (Digital watermark) View project segmentation method based on multi-objective optimization for very high spatial resolution satellite images View project Cloud coverage estimation using ground based images and segmentation techniques.*  
<https://doi.org/10.13140/RG.2.1.4040.9767>
- Murthy, C. B., Hashmi, M. F., Bokde, N. D., & Geem, Z. W. (2020). Investigations of object detection in images/videos using various deep learning techniques and embedded platforms-A comprehensive review. In *Applied Sciences (Switzerland)* (Vol. 10, Issue 9). MDPI AG. <https://doi.org/10.3390/app10093280>
- Naufal, M. F., & Kusuma, S. F. (2022). Weather image classification using convolutional neural network with transfer learning. *AIP Conference Proceedings*, 2470. <https://doi.org/10.1063/5.0080195>
- O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks.*  
<http://arxiv.org/abs/1511.08458>
- Prata, F. (2020). Detection and avoidance of atmospheric aviation hazards using infrared spectroscopic imaging. *Remote Sensing*, 12(14).  
<https://doi.org/10.3390/rs12142309>



- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. <http://arxiv.org/abs/1506.02640>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. <http://arxiv.org/abs/1506.01497>
- Romano, F. (2020). Editorial for the special issue “remote sensing of clouds.” In *Remote Sensing* (Vol. 12, Issue 24, pp. 1–3). MDPI AG. <https://doi.org/10.3390/rs12244085>
- Rossow, W. B., & Schiffer, R. A. (1991). ISCCP Cloud Data Products. In *bulletin of the american meteorological society* (Vol. 72, Issue 1).
- Rudrappa, G., & Vijapur, N. (2020). *Cloud Classification using K-Means Clustering and Content based Image Retrieval Technique*.
- Solawetz, J., & Francesco. (2024). *What is YOLOv8? The Ultimate Guide*.
- Stephens, G. L. (2005). *Cloud Feedbacks in the Climate System: A Critical Review*.
- Sun, X., Liu, L., & Zhao, S. (2011). Whole sky infrared remote sensing of cloud. *Procedia Earth and Planetary Science*, 2(1), 278–283. <https://doi.org/10.1016/j.proeps.2011.09.044>
- Talib, M., Al-Noori, A. H. Y., & Suad, J. (2024). YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection. *Karbala International Journal of Modern Science*, 10(1), 56–68. <https://doi.org/10.33640/2405-609X.3339>
- Tang, Y., Yang, P., Zhou, Z., Chen, J., Pan, D., & Zhao, X. (2020). *Improving Cloud Type Classification of Ground-Based Images Using Region Covariance Descriptors*.
- Taravat, A., Del Frate, F., Cornaro, C., & Vergari, S. (2015). Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters*, 12(3), 666–670. <https://doi.org/10.1109/LGRS.2014.2356616>
- Tyndall, J. (1868). *On the Blue Colour of the Sky, the Polarization of Skylight, and on the Polarization of Light by Cloudy Matter Generally*.
- Wang, Y., & Zhao, C. (2017). Can MODIS cloud fraction fully represent the diurnal and seasonal variations at DOE ARM SGP and Manus sites? *Journal of Geophysical Research: Atmospheres*, 122(1), 329–343. <https://doi.org/10.1002/2016JD025954>

- Xie, W., Liu, D., Yang, M., Chen, S., Wang, B., Wang, Z., Xia, Y., Liu, Y., Wang, Y., & Zhang, C. (2020). SegCloud: A novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation. *Atmospheric Measurement Techniques*, 13(4), 1953–1961. <https://doi.org/10.5194/amt-13-1953-2020>
- Yuchechen, A. E., Gabriela Lakkis, S., Caferri, A., Canziani, P. O., & Muszkats, J. P. (2020). A cluster approach to cloud cover classification over South America and adjacent oceans using a k-means/k-means++ unsupervised algorithm on GOES IR imagery. *Remote Sensing*, 12(18). <https://doi.org/10.3390/RS12182991>
- Zhang, J., Liu, P., Zhang, F., & Song, Q. (2018). CloudNet: Ground-Based Cloud Classification dWith Deep Convolutional Neural Network. *Geophysical Research Letters*, 45(16), 8665–8672. <https://doi.org/10.1029/2018GL077787>
- Zhao, C., Chen, Y., Li, J., Letu, H., Su, Y., Chen, T., & Wu, X. (2019). Fifteen-year statistical analysis of cloud characteristics over China using Terra and Aqua Moderate Resolution Imaging Spectroradiometer observations. *International Journal of Climatology*, 39(5), 2612–2629. <https://doi.org/10.1002/joc.5975>