

Model Convolutional Neural Network yang Efektif dan Efisien untuk Segmentasi Semantik Awan Cumulonimbus

Azminuddin I. S. Azis¹, Jeffry², Firman Aziz³, Andi Taufiqurrahman Akbar⁴

Prodi Teknik Robotika dan Kecerdasan Buatan, Institut Teknologi Bacharuddin Jusuf Habibie¹

Prodi Ilmu Komputer, Institut Teknologi Bacharuddin Jusuf Habibie^{2,4}

Prodi Ilmu Komputer, Universitas Pancasakti Makassar³

Email Korespondensi Author: jeffry@ith.ac.id

This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.



Kata kunci:

Convolutional Neural Network, Segmentasi Semantik Awan Cumulonimbus, Gradient Descent Optimizer, Weighted Class, Data Augmentation.

Abstrak

Awan Cumulonimbus (CB) merupakan jenis awan yang dapat mengakibatkan petir, badai, tornado, hujan lebat, turbulensi penerbangan, dan cuaca ekstrem lainnya. Oleh karenanya, prediksi/deteksi keberadaan awan CB yang akurat dan *real time* akan mendukung kelancaran dan keselamatan banyak aktivitas manusia. Citra infrared (IR) pada satelit Himawari-8 di Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) memiliki informasi mengenai pertumbuhan awan CB. Berbagai studi terkait telah membuktikan bahwa metode yang paling populer dan handal dalam bidang *computer vision* pada objek awan adalah Convolutional Neural Network (CNN). Untuk menemukan/memperoleh model CNN yang paling efektif dan efisien dalam menangani segmentasi semantik awan CB pada citra IR Himawari-8, maka berbagai pendekatan untuk CNN diuji coba, diantaranya arsitektur jaringan untuk CNN, optimalisasi pelatihan CNN berbasis Gradient Descent Optimizer (GDO), Weighted Class (WC) untuk mereduksi masalah *imbalanced class*, dan Data Augmentation (DA) untuk memperkaya keragaman data dan mencegah *overfitting*. Hasil studi menunjukkan bahwa model CNN yang paling efektif adalah dengan arsitektur jaringan U-NET, GDO menggunakan Adaptive Moment Estimation (Adam), dan WC dengan 99,56% global akurasi pengujian, 97,12% rata-rata akurasi pengujian, 94,42% rata-rata IoU, 94,48% akurasi prediksi pada *class* CB, 99,60% akurasi validasi, 99,61% akurasi pelatihan, 0,1071 *loss* validasi, dan 0.1072 *loss* pelatihan. Sedangkan model CNN yang paling efisien adalah dengan arsitektur jaringan Dilated, GDO menggunakan Root Mean Square Propagation (RMSProp), dan WC dengan 24 detik waktu proses/pemodelan, lebih cepat 20 detik namun dengan efektivitas yang tidak jauh berbeda daripada model CNN yang paling efektif.

Keywords:

Convolutional Neural Network, Semantic Segmentation of Cumulonimbus Cloud, Gradient Descent Optimizer, Weighted Class, Data Augmentation.

Abstrack

Cumulonimbus (CB) clouds are a type of cloud that can cause lightning, storms, tornadoes, heavy rain, flight turbulence, and other extreme weather. Therefore, accurate and real-time prediction/detection of CB clouds will support the smoothness and safety of many human activities. Infrared (IR) images on the Himawari-8 satellite at the Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) have information about the growth of CB clouds. Various related studies have proven that the most popular and reliable method in the field of computer vision on cloud objects is the Convolutional Neural Network (CNN). To find/obtain the most effective and efficient CNN model in handling semantic segmentation of CB clouds in Himawari-8 IR images, various approaches for CNN were experimented, including network architecture for CNN, optimization of CNN training based on Gradient Descent Optimizer (GDO), Weighted Class (WC) to reduce imbalanced class problems, and Data Augmentation (DA) to enrich data diversity and prevent overfitting. The results of the study show that the most effective CNN model is with U-NET network architecture, GDO using Adaptive Moment Estimation (Adam), and WC with 99.56% global accuracy in testing, 97.12% mean accuracy in testing, 94.42% mean IoU, 94.48% prediction accuracy on CB class, 99.60% validation accuracy, 99.61% training accuracy, 0.1071 validation loss, and 0.1072 training loss. While the most efficient CNN model is with Dilated network architecture, GDO using Root Mean Square Propagation (RMSProp), and WC with 24 seconds processing/modeling time, 20 seconds faster but with effectiveness that is not much different from the most effective CNN model.

Pendahuluan

Awan memainkan peran penting dalam menentukan perubahan cuaca (Sebastian et al., 2021). Awan *Cumulonimbus* (CB) merupakan jenis awan yang dapat menghasilkan petir, badai, tornado, hujan lebat, dan cuaca ekstrim lainnya (Akbar & Prajitno, 2023; Rozi, 2019; Wanayumini, 2021). Oleh karenanya, keberadaan awan CB dapat membahayakan banyak aktivitas manusia, salah satunya seperti aktivitas penerbangan. Sebagai contoh, Pesawat Air Asia QZ8501 dengan rute penerbangan Surabaya-Singapura dinyatakan hilang sejak Minggu (28/12/2014), karena buruknya cuaca terkait adanya awan CB di jalur penerbangan tersebut (Rudi, 2014). Dengan demikian, informasi dan prediksi/deteksi keberadaan awan CB yang akurat dan *real time* akan mendukung kelancaran dan keselamatan banyak aktivitas manusia, seperti aktivitas penerbangan.

Pendekatan *computer vision* dapat menjadi alternatif yang inovatif untuk mengatasi masalah tersebut, karena lebih sederhana serta tidak membutuhkan biaya dan instrumen yang jauh lebih besar dibandingkan pendekatan konvensional. Selain itu, dengan pendekatan *computer vision*, prediksi awan CB dapat dilakukan secara *real time* dengan memanfaatkan citra satelit awan (Shi & Zuo, 2022). Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) menggunakan satelit dan radar untuk pengindraan jauh terkait keadaan cuaca (BMKG, 2024). Citra *infrared* (IR) dari Satelit Himawari-8 memiliki informasi mengenai suhu titik puncak awan (BMKG, 2024), sehingga pertumbuhan awan CB dapat dilihat pada citra ini (Abidin et al., 2023). Pertumbuhan awan dengan suhu di bawah -60°C dapat dikategorikan sebagai awan konvektif atau CB (Rozi, 2019). Oleh karena itu, citra IR Himawari-8 dapat dimanfaatkan untuk memprediksi/mendeteksi keberadaan awan CB dengan pendekatan *computer vision*.

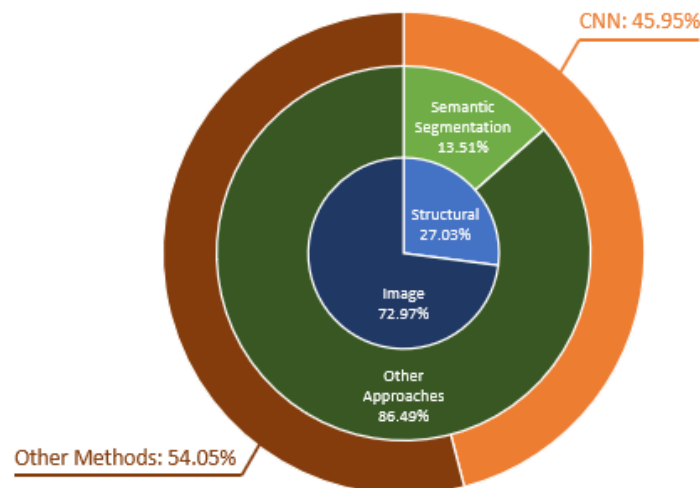
Karena segmentasi objek-objek jenis awan pada citra IR Himawari-8 sangat beragam dan tidak terpisah dengan jelas, maka pendekatan *computer vision* yang paling menjanjikan untuk digunakan adalah segmentasi semantik yang mampu menentukan *label/class* pada setiap piksel dari suatu citra (Brostow et al., 2009; Chen et al., 2018). Sedangkan untuk metodenya, penerapan berbagai metode-metode *machine learning* dan *deep learning* untuk prediksi, klasifikasi, peramalan, maupun segmentasi pada objek/data awan telah menunjukkan kinerja yang tinggi, sebagaimana secara rinci ditunjukkan pada Tabel 1 berikut ini. Berbagai studi terkait tersebut telah membuktikan bahwa metode yang paling populer dan handal dalam bidang *computer vision* pada objek awan adalah *Convolutional Neural Network* (CNN) yang merupakan salah satu varian dari *deep learning*.

Tabel 1. Penelitian Terkait

Referensi	Topik	Type Data	Metode	Kinerja Model
(Lee et al., 1990)	Cloud classification	Structural	ANN	Accuracy = 96%
(Bankert, 1994)	Cloud classification	Image	PNN	Accuracy = 91.2%
(Uddstrom & Gray, 1996)	Cloud classification	Image	MR & MST	Accuracy = 60%
(Liu et al., 2009)	Cloud classification	Image	ANN	Accuracy = 90%
(Q. Wang & Fan, 2010)	CB forecasting	Structural	RS - Artificial Immune	Accuracy = 70%
(Tapakis & Charalambides, 2012)	Cloud classification	Structural	Naïve Bayes	Accuracy = 99.73%
(Jin et al., 2014)	Cloud classification	Image	CCSI-ODSR	Accuracy = 83.2%
(Putra, Angie, Lursinsap, 2014)	CB forecasting	Structural	ANN - PCA	Accuracy = 83.5%
(Cheng & Yu, 2015)	CB initiation	Image	Random Forest	Accuracy = 97.8
(Nugraheny, 2015)	CB classification	Image	PCA	Accuracy = 93%
(Noia & Hasekamp, 2018)	Cloud remote sensing	Structural	ANN - SVM	Good performance
(J. Zhang et al., 2018)	Cloud classification	Image	CNN	Accuracy = 88%
(Cai & Wang, 2018)	Cloud classification	Image	CNN	Accuracy = 90.6%
(Purbantoro et al., 2019)	Cloud classification	Image	SMO	Accuracy = 97.4%
(Sinko et al., 2019)	Cloud classification	Image	CNN	Accuracy = 85%
(Rozi, 2019)	CB prediction	Image	DEEPESN	dB = 22.994
(Segal-Rozenhaimer et al., 2020)	Cloud detection	Image	CNN	Accuracy = 91%
(C. Wang et al., 2020)	Cloud detection	Structural	Random Forest	TPR = 0.93
(Guijo-Rubio et al., 2020)	CB prediction	Structural	Evolutionary ANN	AUC = 90.608%
(Z. Zhang et al., 2020)	Cloud formation	Image	CNN	IoU = 0.82
(Zhao et al., 2020)	Cloud classification	Image	Multi Channel CNN	Accuracy = 94%
(Zhu et al., 2021)	Cloud classification	Image	CNN (ResNet-18)	Accuracy = 91%
(Chattopadhyay, Dutta, et al., 2021)	CB classification	Image	CNN (AlexNet)	Accuracy = 84%
(Chattopadhyay, Pal, et al., 2021)	CB classification	Image	CNN	Accuracy = 81.65%
(Putra et al., 2021)	CB classification	Structural	ANN	Accuracy = 80%
(Sebastian et al., 2021)	Cloud classification	Image	Random Forest	Accuracy = 90%

(Cote, 2022)	Cloud classification	Image	CNN	Better Accuracy
(Nataraja et al., 2022)	Cloud semantic seg.	Image	CNN	Fraction > 80%
(Shi & Zuo, 2022)	Cloud semantic seg.	Image	RCNN	Improves mIoU = 15.24%
(Rais et al., 2022)	CB prediction	Structural	IFS HRES	Better Accuracy
(Juncklaus Martins et al., 2023)	Cloud semantic seg.	Image	CNN	Accuracy = 85%
(Abidin et al., 2023)	CB classification	Image	CNN	Accuracy = 99%
(Omar et al., 2023)	CB classification	Image	CNN	Better Accuracy
(Akbar & Prajitno, 2023)	CB prediction	Structural	ANN-GWO	Accuracy = 92.59%
(Chattopadhyay et al., 2024)	CB classification	Image	CNN	Accuracy = 74%
(Rajendiran et al., 2024)	Cloud segmentation	Image	Machine Learning	Good performance
(Martus & Johnson, 2024)	Cloud segmentation	Image	CNN	Good performance

Studi-studi terkait tersebut menunjukkan bahwa tipe data citra lebih populer digunakan daripada struktural (lihat Gambar 1), menjadi salah satu pula alasan menggunakan pendekatan *computer vision* dalam penelitian ini. Sementara itu, pendekatan segmentasi semantik yang masih jarang diterapkan (lihat Gambar 1), bahkan belum ditemukan penerapannya pada awan CB, menjadi peluang untuk melakukan penelitian pada topik tersebut. Selanjutnya, metode yang paling populer digunakan (lihat Gambar 1) dengan kinerja yang paling handal untuk klasifikasi, prediksi, peramalan, segmentasi, maupun segmentasi semantik pada objek awan adalah CNN. Selain itu, belum ada penelitian-penelitian terkait yang memanfaatkan data citra IR-Himawari-8 untuk segmentasi semantik awan CB berbasis CNN di Indonesia. Dengan demikian, serangkaian uji coba penerapan CNN untuk menemukan model yang paling efektif dan efisien terhadap segmentasi semantik awan CB berdasarkan citra IR Himawari-8 masih terbuka lebar untuk diteliti/dikaji, menjadi kontribusi dan fokus utama dalam penelitian ini.



Gambar 1. Frekuensi Tipe Data, Pendekatan, dan Metode dalam Penelitian terkait Topik/Objek Penelitian (Sumber: Tabel 1)

CNN merupakan metode yang sangat kuat pada bidang *semantic segmentation*, namun lemah terhadap data yang kecil/sedikit, sehingga membutuhkan keragaman data untuk pelatihan modelnya (Nurrani et al., 2023; Yuwana et al., 2020). Selain itu, adanya masalah *imbalanced class* yang pada citra IR Himawari-8 yang terjadi karena probabilitas objek awan CB dalam setiap citra tersebut tentu saja jauh lebih sedikit daripada jenis awan lainnya dan objek-objek lainnya dapat mereduksi kinerja model CNN (Nurrani et al., 2023). Begitupun masalah *overfitting* yang dapat membuat kinerja model CNN buruk pada data yang kondisinya berbeda dengan data latih mesti diatasi (Shen et al., 2018; S. Zhang et al., 2014). Selain itu, agar CNN mampu memberikan kinerja terbaiknya, maka arsitektur jaringannya mesti dirancang sesuai dengan masalah yang ditangani dan dengan metode optimalisasi pelatihannya yang tepat. Masalah-masalah ini, belum ditangani pula secara kolektif dalam penelitian-penelitian terkait sebelumnya.

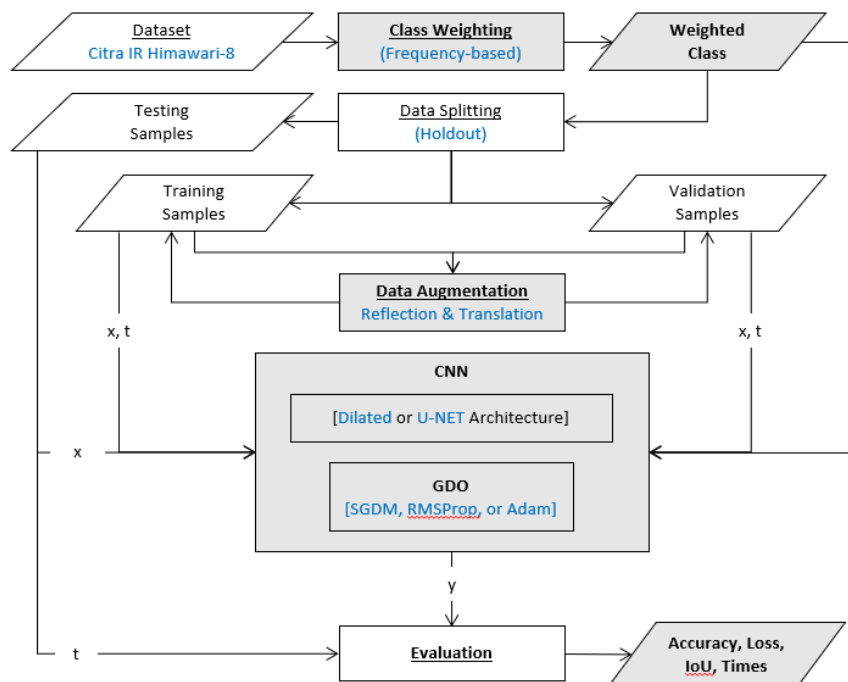
Keragaman data dapat diperkaya dengan menggunakan pendekatan *Data Augmentation* (DA) yang sekaligus dapat pula menangani masalah *overfitting* (Nurrani et al., 2023; Yuwana et al., 2020). Sedangkan masalah *imbalanced class* pada data citra IR Himawari-8 yang terjadi karena probabilitas objek awan CB dalam setiap citra tersebut tentu saja jauh lebih sedikit daripada jenis awan lainnya dan objek-objek lainnya dapat direduksi dengan menggunakan pendekatan *Weighted Class* (WC) (Nurrani et al., 2023). Selanjutnya, arsitektur jaringan yang sederhana/linier yang hanya menekankan fungsi

Dilated hingga yang kompleks seperti U-NET mesti diuji coba untuk menemukan arsitektur jaringan CNN yang terbaik untuk segmentasi semantik awan CB. Begitupun dengan metode optimalisasi pelatihan CNN yang berbasis *Gradient Descent Optimizer* (GDO). Metode berbasis GDO, seperti *Stochastic Gradient Descent with Momentum* (SGDM), *Root Mean Square Propagation* (RMSProp), dan *Adaptive Moment Estimation* (Adam) mesti diuji coba untuk menemukan metode GDO yang terbaik untuk mengoptimalkan pelatihan CNN dalam menangani segmentasi semantik awan CB.

Dengan demikian, penelitian ini bertujuan untuk menemukan/memperoleh model CNN yang paling efektif dan efisien untuk segmentasi semantik awan CB berdasarkan citra IR Himawari-8 dengan cara merancang dan menentukan arsitektur jaringan CNN yang terbaik diantara arsitektur yang sederhana/linier (*Dilated*) hingga yang kompleks (U-NET), menentukan metode GDO (SGDM, RMSProp, atau Adam) yang terbaik untuk optimalisasi pelatihan CNN, menerapkan pendekatan WC dalam lapisan *pixel-classification* CNN untuk mereduksi *imbalanced class*, dan menerapkan pendekatan DA dalam data latih CNN untuk memperkaya keragaman data sekaligus mencegah *overfitting*.

Metode

Penelitian ini menggunakan metode penelitian eksperimen. Alat bantu yang digunakan dalam eksperimen adalah Matlab dengan spesifikasi komputasi (perangkat keras) menggunakan GPU NVIDIA Ge-Force RTX 4060 8 GB, CPU Intel Core i7-14700F 2.10 GHz, RAM 32 GB 6000 MT/s, SSD 1TB M.2 NVMe. Secara umum, metode yang diusulkan untuk segmentasi semantik awan CB berdasarkan citra IR Himawari-8 ditunjukkan pada Gambar 2 berikut ini.

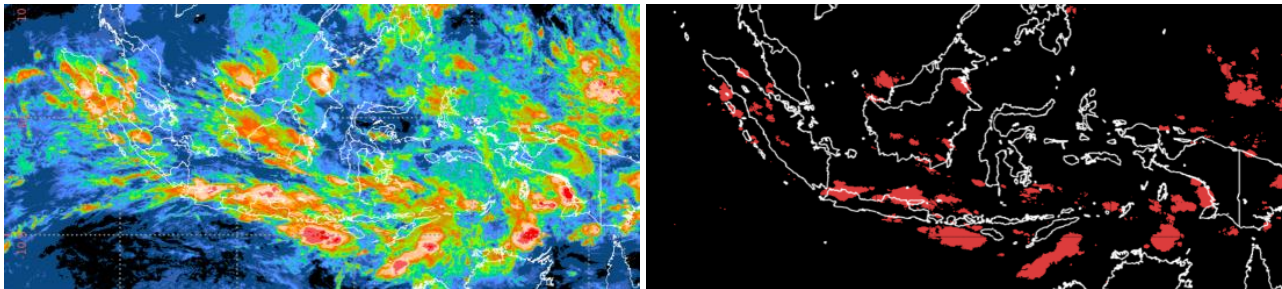


Gambar 2. Metode yang Diusulkan

a. Dataset

Dataset untuk segmentasi semantik awan CB menggunakan citra IR satelit Himawari-8 BMKG. Pengumpulan data ini dilakukan dengan menggunakan teknik dokumentasi yang dikumpulkan dari BMKG. Data citra IR-Enhanced (perhatikan Gambar 3a) merupakan data masukan. Citra tersebut merupakan peningkatan dari citra IR yang diklasifikasi dengan pewarnaan tertentu. Warna hitam atau biru menunjukkan tidak terdapat pembentukan awan yang banyak (cerah). Semakin dingin suhu puncak awan yang ditandai dengan warna mendekati jingga hingga merah menunjukkan pertumbuhan awan yang signifikan dan berpotensi terbentuknya awan CB. Sedangkan citra *convective cloud*

(perhatikan Gambar 3b) merupakan data target. Citra tersebut merupakan produk hasil kolaborasi penelitian BMKG dengan JMA untuk mengidentifikasi secara objektif jenis awan konfektif (tipe awan CB) yang ditangkap oleh band IR dan visibel dari satelit Himawari-8. Warna merah pada citra menunjukkan pertumbuhan awan CB.



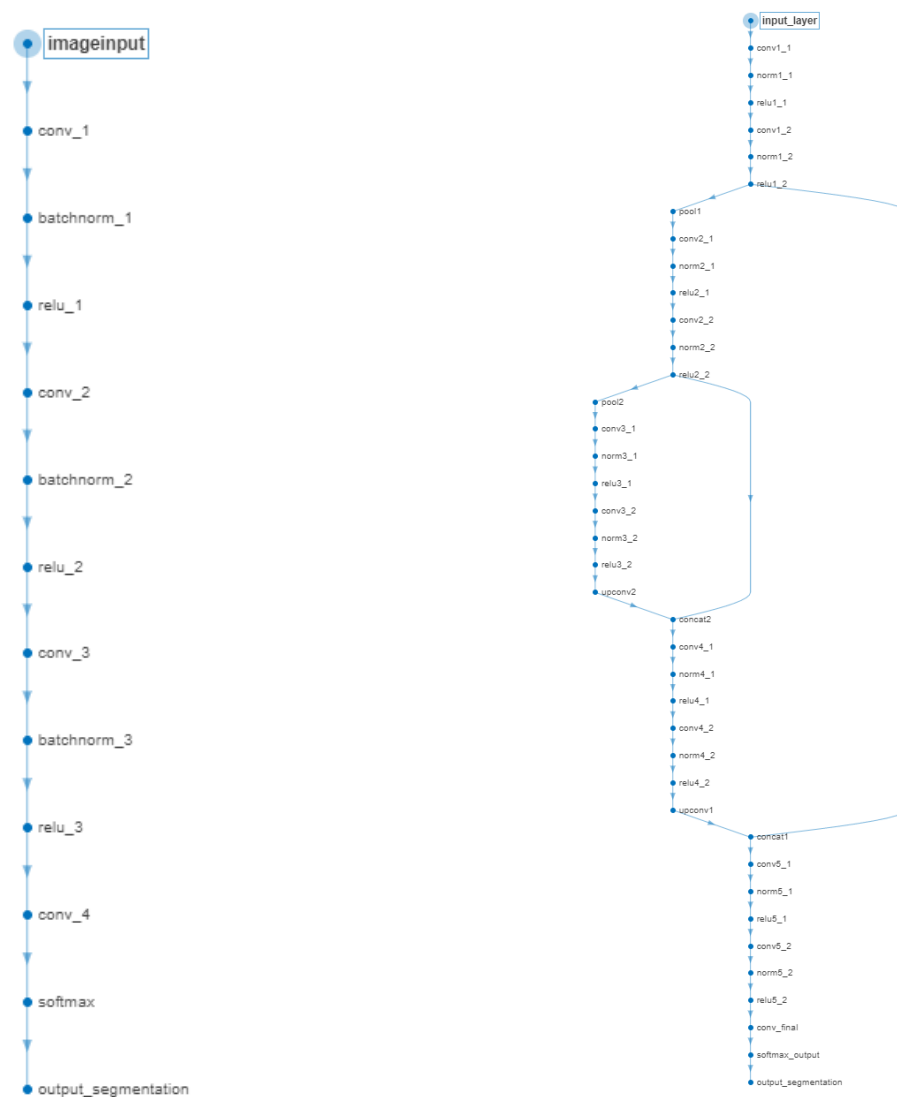
(a) Input

(b) Target

Gambar 3. Sampel Dataset: (a) Citra *IR-Enhanced*; (b) Citra *Convective Cloud*

b. Convolutional Neural Network Architecture

Arsitektur jaringan CNN yang diusulkan untuk menangani segmentasi semantik awan CB adalah arsitektur sederhana/linier yang hanya menekankan pada *Dilated function* sehingga dinamakan *Dilated* (perhatikan Gambar 4a) dan arsitektur bercabang yang kompleks yaitu U-NET (perhatikan Gambar 4b).



(a) Sederhana/Linier

(b) Kompleks/Bercabang

Gambar 4. Arsitektur Jaringan CNN yang Diusulkan: (a) Dilated; (b) U-NET

Secara umum, lapisan-lapisan arsitektur jaringan CNN terdiri dari (Ahlawat et al., 2020):

1. *Input layer*: digunakan untuk memuat data masukan. Argumennya merupakan ukuran tinggi, lebar, dan kanal (1 untuk skala abu-abu dan 3 untuk RGB) dari citra masukan.
2. *Convolutional layer*: digunakan untuk mengekstraksi fitur suatu citra. Sebanyak $n \times n$ neuron masukan pada *input layer* dikonvolusi dengan *filter/kernel* $m \times m$ dan menghasilkan $(n-m+1) \times (n-m+1)$ luaran. Argumen utama lapisan ini adalah *filter size* (kernel), *number of filter* (neurons) *receptive field*, *stride*, *dilation*, dan *padding*.
3. *Batch normalization layer*: digunakan untuk menormalkan aktivasi dan gradien yang merambat melalui jaringan, sehingga pelatihan model CNN menjadi masalah optimasi yang lebih mudah. Lapisan ini digunakan antara *convolutional layer* dan lapisan non-linier (misalnya *ReLU layer*) untuk mempercepat pelatihan model CNN dan mengurangi sensitivitas terhadap inisialisasi jaringan.
4. *ReLU layer*: merupakan lapisan normalisasi *batch* diikuti oleh fungsi aktivasi non-linier. Lapisan ini merupakan lapisan aktivasi dalam *hidden layer*, seperti *softmax* dalam *output layer*.
5. *Pooling layer* atau *down-sampling*: digunakan untuk mengurangi dimensi masukan sehingga mengurangi pula kompleksitas komputasi. *Pooling* memungkinkan nilai-nilai yang dipilih untuk diteruskan ke lapisan berikutnya sambil meninggalkan nilai-nilai yang tidak diperlukan, menghilangkan informasi spasial yang redundan. Lapisan ini juga membantu dalam *feature selection* dan mengendalikan *overfitting*. Tipe *pool* yang umum adalah *maximum pooling* dan *average pooling*. *Maximum pooling* lebih populer karena mengambil nilai maksimum dari setiap sub-wilayah dengan tetap mempertahankan informasi maksimum, sehingga menghasilkan konvergensi yang lebih cepat. Argumen pasangan nilai *stride* menentukan ukuran langkah yang diambil fungsi pelatihan saat memindai masukan.
6. *Fully connected layer*: merupakan lapisan dimana *neuron* terhubung ke semua *neuron* di lapisan sebelumnya. Lapisan ini menggabungkan semua fitur yang dipelajari oleh lapisan sebelumnya untuk mengidentifikasi pola yang lebih besar. *Fully connected layer* yang terakhir menggabungkan fitur-fitur untuk mengklasifikasikan citra. Oleh karena itu, parameter *output size* di *fully connected layer* yang terakhir sama dengan jumlah *class* dalam data target. Lapisan ini memberikan data *flatten-feature* yang merupakan hasil *feature extraction*, sehingga pada dasarnya luaran dari lapisan ini dapat pula dimanfaatkan oleh metode-metode *machine learning* lainnya. Namun untuk segmentasi semantik, lapisan ini tidak bisa digunakan. Lapisan ini berada dalam *output layer*.
7. *Softmax layer*: merupakan fungsi aktivasi *softmax* untuk menormalkan output dari *fully connected layer* atau *final-convolutional layer* (jika segmentasi semantik). Luaran dari lapisan ini terdiri dari bilangan positif yang jumlahnya satu, yang kemudian dapat digunakan sebagai probabilitas klasifikasi oleh *classification layer* atau *pixel-classification layer* (jika segmentasi semantik). Lapisan ini berada dalam *output layer*.
8. *Classification layer*: merupakan lapisan terakhir untuk klasifikasi. Lapisan ini menggunakan probabilitas yang dikembalikan oleh fungsi aktivasi *softmax* untuk setiap masukan untuk menetapkan masukan ke salah satu kelas yang saling eksklusif dan menghitung *loss*. Lapisan ini memberikan hasil prediksi dari masukan. Pada segmentasi semantik, lapisan ini menggunakan istilah *pixel-classification layer*.

c. Convolutional Neural Network Options

Secara umum, berbagai pengaturan untuk model CNN terdiri dari:

- *Learning rate schedule*: menurunkan *learning rate* (LR) seiring waktu untuk meningkatkan stabilitas dan akurasi di akhir pelatihan. Nilai default = none, terbaik = *piecewise*.
- *Learning rate drop period*: menurunkan LR setiap x *epoch* jika *learning rate schedule* = *piecewise*. Nilai default = 10.
- *Learning rate drop factor*: LR baru = LR lama $\times x$. Menurunkan LR. Nilai default = 10.
- Momentum: dapat mempercepat pelatihan dan kondisi konvergensi lebih jauh tetapi meningkatkan resiko *overshooting* (melewati) minima yang dicari, sehingga bisa jadi malah mereduksi akurasi pula. Nilai 0,9 (default) adalah nilai yang baik dapat memberikan percepatan konvergensi tanpa menyebabkan *overshooting*. Opsi ini hanya jika menggunakan metode GDO = SGDM.
- *Gradient decay factor* (beta1): mengarahkan *optimizer* ke arah gradien yang konsisten. Nilai 0.9 (default dan terbaik) berarti estimasi saat ini dipengaruhi oleh 90% dari rata-rata *gradien*

- sebelumnya. Opsi ini mengontrol kecepatan pergerakan *optimizer* melalui permukaan *loss*. Nilai yang lebih tinggi (mendekati 1) memberikan lebih banyak bobot pada *gradien* sebelumnya (meningkatkan momentum). Opsi ini hanya jika menggunakan metode GDO = Adam.
- *Squared gradient decay factor* (beta2): mengukur variabilitas (sebaran) *gradien*. Informasi ini digunakan untuk menyesuaikan LR secara individual untuk setiap bobot/parameter. Opsi ini hanya jika menggunakan metode GDO = Adam atau RMSProp, memungkinkan Adam menjadi adaptif. Parameter dengan *gradien* yang sangat bervariasi (cepat berubah) akan diberi LR yang lebih kecil, dan sebaliknya. Nilai ini harus sangat dekat dengan 1 untuk memberikan estimasi variabilitas yang stabil pada setiap parameter. Nilai terbaik dan default = 0.999.
 - *Epsilon*: nilai kecil untuk memastikan bahwa pembagian dengan nol tidak pernah terjadi. Menjaga stabilitas, dimana jika variabilitas gradien ini mendekati nol, maka pembagian akan menghasilkan nilai yang sangat besar (*infinity*), menyebabkan bobot meledak (*exploding weights*) dan membuat pelatihan tidak stabil. Opsi ini hanya jika menggunakan metode GDO = Adam atau RMSProp. Nilai terbaik = $1e-8$.
 - *Gradient threshold*: merupakan teknik regularisasi yang disebut *clipping gradient*. Fungsi utamanya adalah untuk mencegah bobot jaringan berubah secara drastis dalam satu iterasi. Penting digunakan pada RNN atau jika *mini batch size* kecil. Nilai yang konservatif = [1, 5].
 - *Gradient threshold method*: merupakan metode untuk *gradient threshold*. Nilainya = [*l2norm* (terbaik), *absolut-value*, *global-l2norm*].
 - *Initial learning rate*: minimum error. Nilai default = $1e-2$, terbaik = $1e-3$.
 - *L2 regularization*: opsi ini sangat penting karena dapat mencegah *overfitting* dan meningkatkan generalisasi (menyederhanakan model). Nilai default = $1e-4$.
 - *Validation data*: data untuk validasi.
 - *Max epochs*: iterasi maksimum. Nilai default = 30.
 - *Mini batch size*: jumlah sample (gambar/label) yang diproses dalam satu iterasi sebelum *optimizer* memperbarui bobot jaringan. Nilainya sesuaikan memori GPU [8, 16, ...].
 - *Shuffle*: menentukan apakah data pelatihan akan diacak di awal setiap *epoch* untuk mencegah jaringan mempelajari urutan data. Nilai default = *once*, terbaik = *every-epoch*.
 - *Check point path*: menentukan *folder* (misalnya *tempdir*) untuk menyimpan *checkpoint* (titik pemulihan) jaringan selama pelatihan, memungkinkan pelatihan dilanjutkan setelah gangguan.
 - *Plots*: proses pelatihan diplot secara visual dalam jendela terpisah atau tidak.
 - *Validation patience*: menentukan batas jumlah *epoch* dimana validasi *loss* tidak membaik sebelum pelatihan dihentikan secara otomatis (*early stopping*). Berhenti jika *loss* tidak membaik selama x *epoch*. Nilai default = inf, terbaik = [5, 10].
 - *Validation frequency*: memeriksa validasi setiap x iterasi. Nilai default = 50.
 - *Verbose frequency*: menampilkan hasil setiap x iterasi. Nilai default = 50.
 - *Verbose*: menampilkan informasi pelatihan di jendela atau tidak.
 - *Execution environment*: Menggunakan sumber daya gpu, cpu, multi gpu, atau auto. Sesuaikan dengan sumber daya yang digunakan.

d. Gradient Discent Optimization

Gradient Discent Optimizer (GDO) merupakan salah satu pendekatan optimasi yang dapat digunakan dalam pelatihan CNN untuk menghasilkan kinerja yang lebih baik dan hasil yang lebih cepat. Pendekatan ini membantu meminimalkan atau memaksimalkan *cost function* dengan memperbarui nilai bobot/bias secara adaptif (Ahlawat et al., 2020). Sesuai namanya, pendekatan ini menggunakan *error gradien* untuk turun bersama dengan *error surface*. Hal ini juga memungkinkan nilai minimum suatu fungsi ditemukan ketika turunannya ada dan hanya ada satu solusi optimal (jika mengharapkan nilai minimum lokal). Gradien adalah kemiringan *error surface* dan memberikan indikasi seberapa sensitif *error* terhadap perubahan bobot (Ahlawat et al., 2020). Sensitivitas ini dapat dimanfaatkan untuk perubahan bobot secara bertahap menuju nilai optimal. Pendekatan ini menghindari komputasi redundan yang lebih baik daripada pendekatan lainnya. Pendekatan ini bekerja lebih cepat dan telah banyak digunakan dalam eksperimen CNN (Ahlawat et al., 2020). Metode GDO yang umum digunakan pada CNN adalah *Stochastic Gradient Descent with Momentum* (SGDM), *Root Mean Square Propagation* (RMSProp), dan

Adaptive Moment Estimation (Adam) (Ahlawat et al., 2020). Algoritma pendekatan GDO ditunjukkan pada Gambar 5 berikut ini.

- 1: **Input:** $(x(i), y(i))$ as training data; η as learning rate; $\nabla E(\theta)$ is a gradient of loss (error) function $E(\theta)$ with respect to the θ parameter; momentum factor (m)
- 2: **Output:** For each training pair, update the learning parameters using the equation

$$\theta = \theta - \eta \cdot \nabla E(\theta; x(i); y(i))$$
- 3: if stopping condition is met
- 4: return parameter θ .

Gambar 5. Algoritma Pendekatan GDO (Ahlawat et al., 2020)

e. Weighted Class

Pendekatan *Weighted Class* (WC) dapat digunakan untuk mereduksi *imbalanced class* yang terjadi pada data, sehingga model CNN dalam pelatihannya tidak lebih cenderung memprediksi output ke *class/label* yang lebih banyak komposisinya sebagaimana dataset citra IR Himawari-8 menunjukkan gejala tersebut. Tujuan pendekatan ini memberikan bobot yang lebih besar terhadap *class* yang lebih sedikit frekuensinya. Selanjutnya bobot tersebut dapat diterapkan dalam *classification layer* CNN. Namun perlu diperhatikan pula perbedaan bobot yang signifikan/ekstrem antar *class* akan membuat model CNN terlalu agresif sehingga justru menurunkan kinerjanya. Adapun formulasi WC yang diusulkan ditunjukkan pada Persamaan (1) berikut ini.

$$w_i = \frac{\text{median}(f_i)}{f_i} \quad (1)$$

Dimana w adalah bobot untuk setiap *class i* dan f adalah frekuensi untuk setiap *class i*.

Jika selisih bobot yang diberikan antara *class_min* dan *class_max* terlalu besar/ekstrem, maka *class i* dengan bobot terkecil (*class_min*) diberikan bobot = 1, sedangkan *class i* yang satunya (*class_max*) diberikan bobot dengan menggunakan Persamaan (2) berikut ini.

$$w_{\text{class_max}} = \text{median} \left(\left| \frac{\sqrt{w_{\text{class_max}}/w_{\text{class_min}}}}{2} \right| w_{\text{class_min}} \right) \quad (2)$$

f. Data Augmentation

Pendekatan *Data Augmentation* (DA) digunakan untuk memperkaya keragaman data sekaligus mencegah *overfitting*. dengan menambahkan data latih menggunakan pendekatan *random reflection* dan *translation*, dimana nilai $x\text{Translation} = [-30, 30]$ dan $y\text{Translation} = [-30, 30]$.

g. Model Evaluation and Comparison

Pendekatan yang digunakan untuk mengukur kinerja akurasi dan *Intersection over Union* (IoU) dari model-model yang diuji coba adalah *Confusion Matrix*, ditunjukkan pada Tabel 2 berikut ini.

Tabel 2, Confusion Matrix

Label Class		Predicted (Output)	
		P	N
Actual (Target)	P	TP	FN
	N	FP	TN

Berdasarkan tabel *Confusion Matrix*, maka kinerja global akurasi (gA) dihitung menggunakan Persamaan (3), rata-rata akurasi (mA) dihitung menggunakan Persamaan (5) berdasarkan akurasi setiap *class i* (A_i) atau *recall* yang dihitung menggunakan Persamaan (4), dan rata-rata IoU ($mIoU$) dihitung menggunakan Persamaan (7) berdasarkan IoU setiap *class i* (IoU_i) yang dihitung menggunakan Persamaan (6).

$$gA = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$A_i = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

$$mA = \frac{1}{n} \sum_{i=1}^n A_i \quad (5)$$

$$IoU_i = \frac{Area\ Output \cap Area\ Target}{Area\ Output \cup Area\ Target} = \frac{TP_i}{TP_i + FP_i + FN_i} \quad (6)$$

$$mIoU = \frac{1}{n} \sum_{i=1}^n IoU_i \quad (7)$$

Sedangkang kinerja *loss* dihitung menggunakan pendekatan *Binnary Cross-Entropy Loss*, dimana formulasinya per piksel dihitung menggunakan Persamaan (8) berikut ini.

$$\mathcal{LCE} = -[y \log(\bar{y}) + (1 - y) \log(1 - \bar{y})] \quad (8)$$

Dimana y adalah label *ground truth* piksel (1 untuk *class CB*, dan 0 untuk *class Others*), sedangkan $\bar{y}$ adalah probabilitas piksel doprediksi sebagai *class CB*.

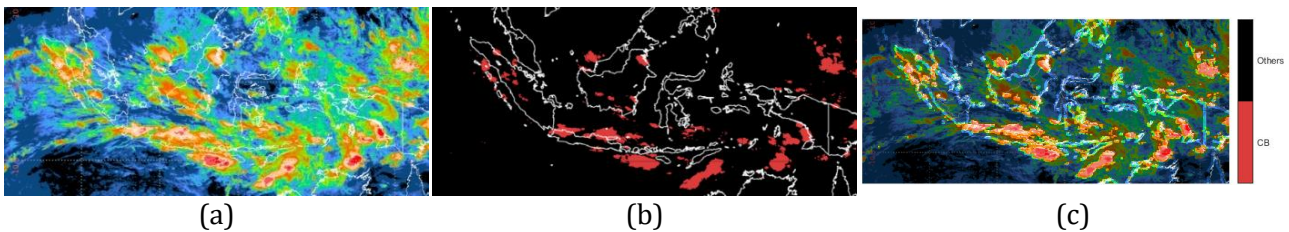
Jika menggunakan WC (Wc), maka total *loss* adalah rata-rata *loss* disemua piksel (n), dengan bobot yang dihitung menggunakan Persamaan (9) berikut ini.

$$\mathcal{L}_{weightedCE} = \frac{1}{n} \sum_{i=1}^n Wc_i \cdot [y_i \log(\bar{y}_i) + (1 - y_i) \log(1 - \bar{y}_i)] \quad (9)$$

Dimana Wc_i adalah bobot *class* yang sesuai dengan piksel i .

Hasil dan Diskusi

Sampel penggabungan antara citra input dan target beserta class pikselnya ditunjukkan pada Gambar 5 berikut ini.



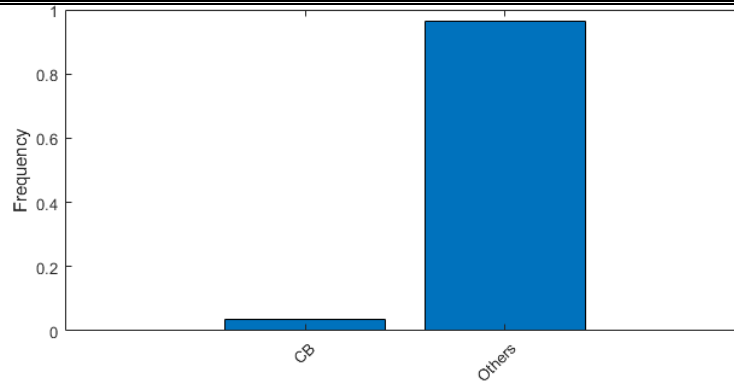
Gambar 5. Sampel Penggabungan Citra: (a) Citra Input; (b) Citra Target; (c) Citra Gabungan

Jumlah piksel setiap *class* (*CB* dan *Others*) pada seluruh data target ditunjukkan pada Tabel 3 berikut ini.

Tabel 3. Jumlah Piksel setiap *Class* pada Data Target

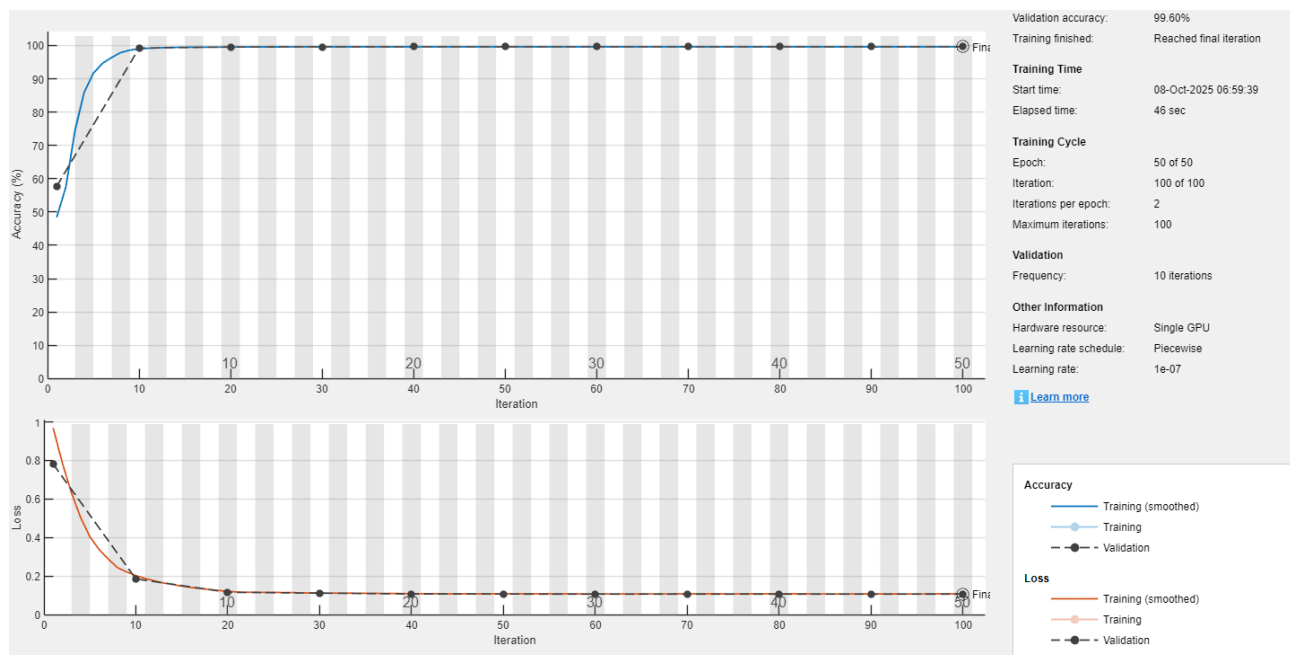
Class	PixelCount	ImagePixelCount
CB	101458	3342800
Others	2679412	3342800

Selanjutnya frekuensi setiap *class* ditunjukkan pada Gambar 6 berikut ini. Hal ini menunjukkan bahwa terjadi *imbalanced class* pada data. Masalah ini akan ditangani dengan menggunakan pendekatan WC.



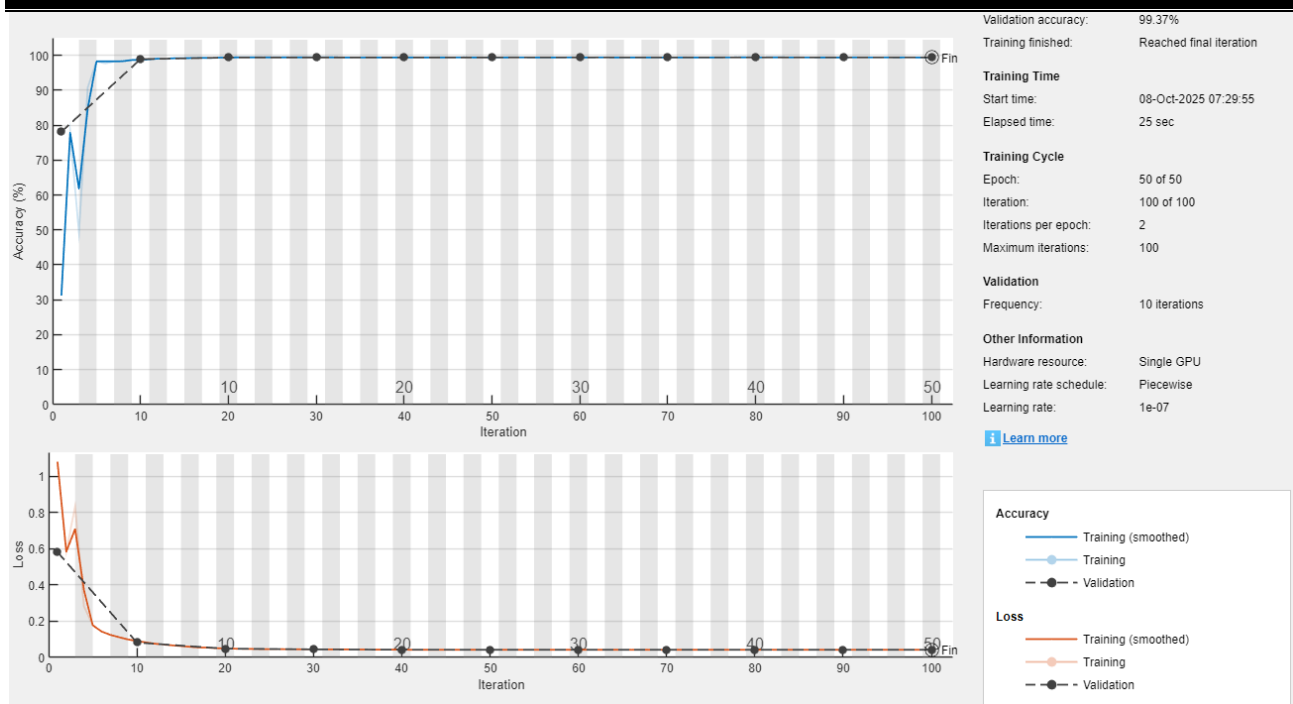
Gambar 6. Diagram Frekuensi setiap Class

Hasil eksperimen menunjukkan bahwa model CNN yang paling efektif untuk segmentasi semantik awan CB pada citra IR Himawari-8 adalah dengan menggunakan arsitektur jaringan U-NET, GDO menggunakan metode Adam, dan dengan menerapkan pendekatan WC memberikan kinerja global akurasi di pengujian sebesar 99,56%, rata-rata akurasi pengujian sebesar 97,12%, rata-rata IoU sebesar 94,42%, akurasi prediksi pada class CB sebesar 94,48%, akurasi validasi sebesar 99,60%, akurasi pelatihan sebesar 99,61%, loss validasi sebesar 0,1071, dan loss pelatihan sebesar 0.1072. Hasil pemodelan dari model ini memberikan kinerja pelatihan dan validasi yang sangat baik dan seimbang. Model ini mampu belajar dengan sangat baik dan juga mampu memberikan kinerja prediksi yang sangat tinggi. Hal ini mengindikasikan tidak ada gejala *overfitting* sehingga model ini sudah general dan sangat kuat digunakan untuk memprediksi data yang baru/berbeda. Begitupun model ini tidak mengindikasikan terjadinya gejala *underfitting* (model yang buruk). Itulah mengapa penerapan DA tidak dapat menunjukkan kinerja yang lebih baik daripada tidak menggunakannya pada model ini. Proses pemodelan model ini ditunjukkan pada Gambar 7 berikut ini.



Gambar 7. Proses Pelatihan dan Evaluasi Model CNN terbaik dari sisi Efektivitas

Sedangkan model CNN yang paling efisien untuk segmentasi semantik awan CB pada citra IR Himawari-8 adalah dengan menggunakan arsitektur jaringan *Dilated*, GDO menggunakan metode RMSProp, dan dengan menerapkan pendekatan WC memberikan waktu proses/pemodelan yang sangat cepat, yaitu 24 detik, lebih cepat 20 detik namun dengan efektivitas yang tidak jauh berbeda daripada model CNN yang paling efektif. Proses pemodelan model ini ditunjukkan pada Gambar 8.



Gambar 8. Proses Pelatihan dan Evaluasi Model CNN terbaik dari sisi Efisiensi

Hasil validasi dan pengujian serta komparasi berbagai model CNN yang diuji coba ditunjukkan pada Tabel 4 berikut ini.

Tabel 4 Komparasi Kinerja seluruh Model

Method	Training		Validation		Global		Mean		Testing		Times (seconds)	No Overfitting	No Underfitting	No Unbalance	CB Prediction
	Accuracy	Loss	Accuracy	Loss	Accuracy	Accuracy	Mean IoU	Mean IoU	Accuracy-CB	Accuracy- Others					
SGDM + Dilated	83,88%	0,3572	83,41%	0,3579	84,07%	88,51%	50,97%	93,32%	83,68%	18,46%	83,47%	✓	✓	✗	✓
SGDM + U-NET	90,38%	0,3218	90,11%	0,3119	90,27%	92,73%	58,68%	95,41%	90,06%	27,48%	89,89%	✓	✓	✗	✓
RMSProp + Dilated	99,42%	0,0298	99,40%	0,0299	99,36%	95,29%	91,97%	90,87%	99,70%	84,61%	99,34%	✓	✓	✓	✓
RMSProp + U-NET	99,48%	0,0243	99,48%	0,0243	99,46%	95,77%	93,07%	91,78%	99,76%	86,71%	99,44%	✓	✓	✓	✓
Adam + Dilated	96,28%	0,221	96,14%	0,2199	96,19%	97,12%	72,96%	98,13%	96,11%	49,88%	96,04%	✓	✓	✓	✓
Adam + U-NET	99,60%	0,0905	99,60%	0,0902	99,56%	96,51%	94,36%	93,21%	99,82%	89,18%	90,55%	✓	✓	✓	✓
SGDM + Dilated + WC	87,58%	0,3732	87,15%	0,3735	87,81%	91,46%	55,29%	95,41%	87,51%	23,23%	87,34%	✓	✓	✗	✓
SGDM + U-NET + WC	92,60%	0,3236	92,34%	0,3237	92,48%	93,66%	62,49%	94,94%	92,38%	32,79%	92,19%	✓	✓	✓	✓
RMSProp + Dilated + WC	99,40%	0,0405	99,37%	0,041	99,35%	96,37%	92,01%	93,15%	99,60%	84,70%	99,33%	✓	✓	✓	✓
RMSProp + U-NET + WC	99,50%	0,0336	99,48%	0,0338	99,47%	96,79%	93,31%	93,89%	99,69%	87,18%	99,45%	✓	✓	✓	✓
Adam + Dilated + WC	96,20%	0,2612	96,07%	0,2598	96,11%	97,14%	72,68%	98,25%	96,02%	49,40%	95,96%	✓	✓	✓	✓
Adam + U-NET + WC	99,61%	0,1072	99,60%	0,1071	99,56%	97,12%	94,42%	94,48%	99,77%	89,29%	99,55%	✓	✓	✓	✓
SGDM + Dilated + DA	83,42%	0,3456	82,50%	0,4853	83,29%	82,74%	49,36%	82,13%	83,34%	15,97%	82,75%	✓	✓	✗	✓
SGDM + U-NET + DA	87,47%	0,3083	85,70%	0,4807	88,28%	88,86%	53,15%	91,67%	86,06%	20,52%	85,77%	✓	✓	✗	✓
RMSProp + Dilated + DA	98,46%	0,0425	98,53%	0,0478	98,48%	88,59%	82,46%	77,87%	99,31%	66,48%	98,44%	✓	✓	✓	✓
RMSProp + U-NET + DA	98,63%	0,0366	98,58%	0,0461	98,55%	89,05%	83,12%	78,76%	99,35%	67,73%	98,50%	✓	✓	✓	✓
Adam + Dilated + DA	91,35%	0,2689	91,92%	0,4461	92,22%	93,57%	61,99%	95,04%	92,10%	32,06%	91,92%	✓	✓	✓	✓
Adam + U-NET + DA	98,65%	0,1034	98,57%	0,1311	98,54%	84,41%	81,59%	69,09%	99,37%	64,68%	98,50%	✓	✓	✓	✓
SGDM + Dilated + WC + DA	86,90%	0,3812	86,46%	0,4862	87,21%	87,14%	53,80%	97,06%	87,22%	20,80%	86,77%	✓	✓	✗	✓
SGDM + U-NET + WC + DA	88,93%	0,3321	88,80%	0,4432	89,25%	90,75%	56,89%	93,37%	89,13%	24,93%	88,85%	✓	✓	✓	✓
RMSProp + Dilated + WC + DA	98,30%	0,0684	98,33%	0,0711	98,27%	90,35%	81,44%	81,76%	98,93%	64,60%	98,21%	✓	✓	✓	✓
RMSProp + U-NET + WC + DA	98,47%	0,0614	98,43%	0,0666	98,42%	91,29%	82,77%	83,56%	99,02%	67,17%	98,37%	✓	✓	✓	✓
Adam + Dilated + WC + DA	91,16%	0,3488	91,75%	0,5112	92,04%	93,67%	61,70%	95,43%	91,90%	31,66%	91,73%	✓	✓	✓	✓
Adam + U-NET + WC + DA	98,47%	0,0614	98,44%	0,0666	98,42%	91,29%	82,79%	83,56%	99,02%	67,21%	98,37%	✓	✓	✓	✓

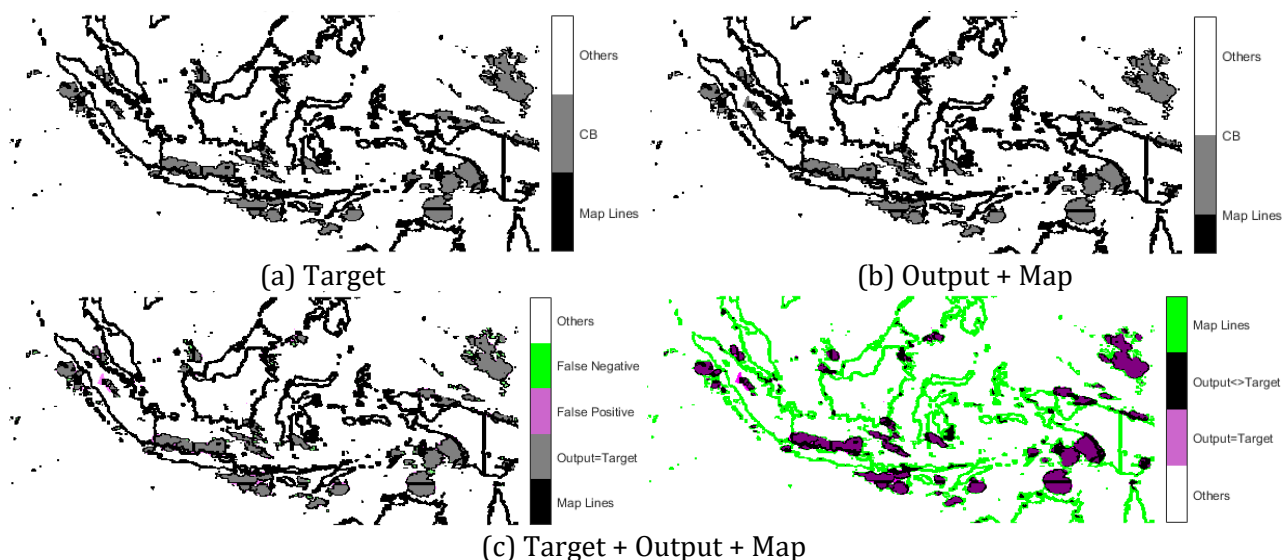
Berdasarkan hasil validasi dan pengujian, maka dapat ditarik kesimpulan bahwa penerapan DA tidak dapat meningkatkan kinerja model CNN. Hal ini mengindikasikan bahwa data yang digunakan sudah cukup mewakili dan juga pada seluruh model CNN yang diuji coba tidak menunjukkan gejala *overfitting*. Sedangkan pendekatan WC sedikit mampu meningkatkan kinerja model CNN karena penyesuaian pembobotan yang tepat untuk kedua *class*, agar tidak terjadi distorsi *loss function*, fokus berlebihan pada *recall*, dan peningkatan *False Positives*, walaupun terjadi *imbalanced class* yang sangat besar. Sementara metode untuk GDO, rata-rata RMSProp dan Adam memberikan kinerja yang tinggi, konsisten, dan seimbang. Namun tidak dengan SGDM, metode ini di beberapa banyak model menunjukkan ketidakseimbangan prediksi, bahkan sedikit mengindikasikan ketidakmampuan dalam memprediksi *class* CB, dan menghasilkan model yang agak buruk (mengindikasikan adanya gejala *underfitting*). Selanjutnya untuk arsitektur jaringan CNN, arsitektur *Dilated* yang sederhana dan linier tidak dapat mengungguli kinerja efektivitas arsitektur U-NET, namun tidak signifikan berbeda. Walaupun demikian, arsitektur *Dilated* memberikan kinerja efisiensi (waktu pemodelan) yang selalu lebih baik daripada

arsitektur U-NET yang jauh lebih kompleks. Sehingga kinerja efisiensi model CNN hanya tergantung pada arsitektur yang digunakan, semakin kompleks arsitekturnya maka semakin buruk pula kompleksitas komputasinya terhadap waktu, walaupun tidak berbeda pula secara signifikan karena tidak sampai selisih menit. Perlu diperhatikan juga bahwa arsitektur U-NET dirancang untuk mereduksi kompleksitas komputasi terhadap ruang dengan adanya *pooling layer* yang dapat diterapkan, sedangkan pada arsitektur *Dilated* tidak dapat menerapkannya karena *pooling layer* pada masalah segmentasi semantik memang tidak dapat diterapkan secara langsung. Pengukuran kompleksitas komputasi terhadap ruang ini bisa menjadi studi selanjutnya untuk lebih menemukan model CNN yang efisien.

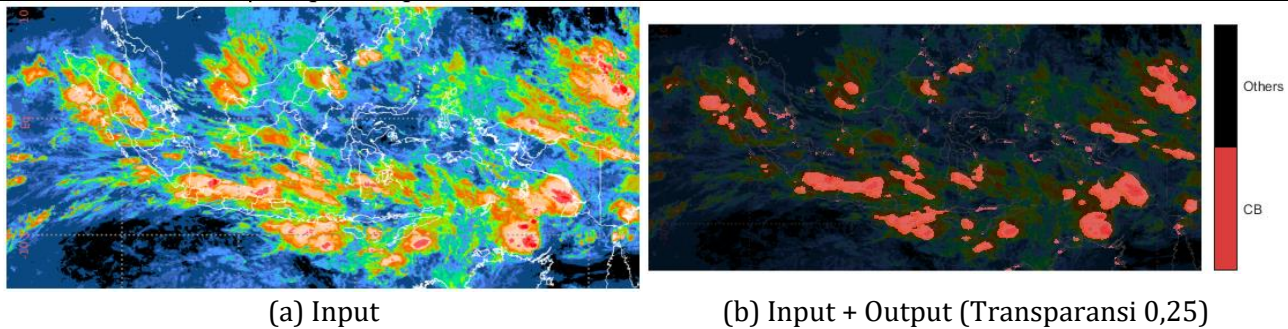
Metriks kinerja yang paling utama adalah mIoU (rata-rata IoU), lebih utama daripada global akurasi. Karena metriks ini memperlakukan semua *class* setara dan menghukum prediksi yang salah secara geometris. Berbeda dengan global akurasi yang bisa *misleading*, hasil prediksi lebih didominasi oleh *class* mayoritas. Bahkan rata-rata akurasi masih lebih utama daripada global akurasi, walaupun tidak terlalu sensitif terhadap kesalahan *False Positive* (memprediksi *class* Others). Begitupun dalam mengukur model CNN yang mana yang paling efektif dalam memprediksi *class* CB, metriks IoU *class* CB adalah yang paling utama, daripada akurasi *class* CB. Sementara itu, kesenjangan yang besar antara kinerja di validasi/pengujian dan pelatihan menunjukkan terjadinya *overfitting*, namun hal ini tidak terjadi pada seluruh model CNN yang diuji coba. Sedangkan kinerja (baik akurasi maupun *loss*) yang buruk di pelatihan, validasi, dan pengujian mengindikasikan terjadinya *underfitting*. Namun hal ini tidak terjadi, hanya sedikit gejala pada model CNN yang menggunakan metode SGDM untuk GDO.

Dibandingkan model CNN lainnya, model CNN berbasis Adam + U-NET + WC memenangkan kinerja mIoU di pengujian (94,42%) sehingga sangat ideal/seimbang dan akurat, IoU *class* CB (89,29%) sehingga sangat bisa digunakan untuk memprediksi awan CB (walaupun *class* CB sangat minoritas), global akurasi di pengujian (99,56%) sehingga sangat akurat, dan akurasi di pelatihan (99,61%) yang sejalan dengan akurasi di validasi (99,60%) menunjukkan tidak adanya *overfitting* terlebih *underfitting*. Kinerjanya membuat model ini sangat efektif, walaupun tidak lebih efisien/efektif dari sisi waktu karena arsitektur jaringannya yang lebih kompleks.

Simulasi prediksi model CNN berbasis Adam + U-NET + WC pada sebuah citra inputan baru beserta targetnya ditunjukkan pada Gambar 9 dan Gambar 10 berikut ini.



Gambar 9. Simulasi Prediksi: Target (*Actual Output*) dan Output (*Predicted Output*)



Gambar 10. Simulasi Prediksi: Input dan Output

Kesimpulan

Studi ini mengusulkan model CNN yang paling efektif dan efisien untuk segmentasi semantik awan CB pada citra IR satelit Himawari-8. Hasil eksperimen menunjukkan bahwa model CNN yang paling efektif adalah dengan arsitektur jaringan U-NET, GDO menggunakan *Adaptive Moment Estimation* (Adam), dan WC dengan 99,56% global akurasi, 97,12% rata-rata akurasi, 94,42% mIoU, 94,48% akurasi prediksi pada *class* CB, 99,60% akurasi validasi, 99,61% akurasi pelatihan, 0,1071 *loss* validasi, dan 0.1072 *loss* pelatihan. Sedangkan model CNN yang paling efisien adalah dengan arsitektur jaringan *Dilated*, GDO menggunakan *Root Mean Square Propagation* (RMSProp), dan WC dengan 24 detik waktu proses/pemodelan, lebih cepat 20 detik namun dengan efektivitas yang tidak jauh berbeda daripada model CNN yang paling efektif. Dengan demikian, penerapan WC untuk mereduksi *imbalanced class* mampu meningkatkan kinerja CNN. Namun penerapan DA untuk memperkaya keragaman data sekaligus mencegah *overfitting* tidak berhasil meningkatkan kinerja CNN karena kinerja model CNN yang sudah sangat efektif/tinggi dan stagnan dengan pendekatan-pendekatan lainnya. Sementara itu, metode GDO yang direkomendasikan untuk optimalisasi pelatihan CNN adalah Adam dan RMSProp. Sedangkan arsitektur jaringan CNN yang unggul dari sisi efektivitas adalah U-NET dan dari sisi efisiensi/waktu adalah *Dilated*.

Namun perlu diperhatikan bahwa arsitektur U-NET dirancang untuk mereduksi kompleksitas komputasi terhadap ruang dengan adanya *pooling layer* yang dapat diterapkan, sedangkan pada arsitektur *Dilated* tidak dapat menerapkannya karena *pooling layer* pada masalah segmentasi semantik memang tidak dapat diterapkan secara langsung. Dengan demikian, pengukuran kompleksitas komputasi terhadap ruang ini bisa menjadi studi selanjutnya untuk lebih menemukan model CNN yang efisien sekaligus efektif. Selain itu, penerapan metode GDO yang lainnya patut diuji coba pada penelitian selanjutnya.

Referensi

- Abidin, M. R., Dian candra Rini Novitasari, Hani Khaulasari, & Fajar Setiawan. (2023). Classification of Cumulonimbus Cloud Formation based on Himawari Images using Convolutional Neural Network Model GoogleNet. *Jurnal Buana Informatika*, 14(2), 127–136. <https://doi.org/10.24002/jbi.v14i02.7417>
- Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved Handwritten Digit Recognition Using Convolutional Neural Network (CNN). *Sensors*, 3344(20), 1–18. <https://doi.org/10.1504/IJCSM.2023.131625>
- Akbar, A., & Prajitno, P. (2023). Cumulonimbus cloud prediction using Artificial Neural Network (ANN) based on Grey Wolf Optimizer (GWO). *THE 1ST INTERNATIONAL CONFERENCE ON SCIENCE EDUCATION AND SCIENCES*. <https://doi.org/10.1063/5.0122544>
- Bankert, R. L. (1994). Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *Journal of Applied Meteorology*, 33(8), 909–918. [https://doi.org/10.1175/1520-0450\(1994\)033<0909:CCOAI>2.0.CO;2](https://doi.org/10.1175/1520-0450(1994)033<0909:CCOAI>2.0.CO;2)
- BMKG. (2024). *Citra Satelit Himawari-8*. Badan Meteorologi, Klimatologi, Dan Geofisika (BMKG). <https://www.bmkg.go.id/satelit/>

- Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2), 88–97. <https://doi.org/10.1016/j.patrec.2008.04.005>
- Cai, K., & Wang, H. (2018). Cloud Classification of Satellite Image based on Convolutional Neural Networks. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 874–877. <https://doi.org/10.1109/ICSESS.2017.8343049>
- Chattopadhyay, S., Chowdhury, A., Dutta, S. P., Mukhopadhyay, K., Ghosh, J., & De, P. (2024). Cumulonimbus cloud classification using transfer learning with AlexNet. *FOURTH INTERNATIONAL CONFERENCE ON ADVANCES IN PHYSICAL SCIENCES AND MATERIALS (ICAPSM)*. <https://doi.org/https://doi.org/10.1063/5.0216143>
- Chattopadhyay, S., Dutta, S. P., Mishra, S., & Saha, S. (2021). Use of Marker-Controlled Watershed Segmentation to Classify Cumulonimbus Cloud with Pre-trained CNN. *Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing*. https://doi.org/https://doi.org/10.1007/978-981-16-1696-9_6
- Chattopadhyay, S., Pal, S., Acharjya, P. P., & Bhattacharyya, S. (2021). A Comparative Study to Classify Cumulonimbus Cloud Using Pre-trained CNN. In: Kumar R., Quang N.H., Kumar Solanki V., Cardona M., Pattnaik P.K. (Eds) *Research in Intelligent and Computing in Engineering. Advances in Intelligent Systems and Computing*. https://doi.org/https://doi.org/10.1007/978-981-15-7527-3_11
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic. *ECCV*, 833–851. <https://doi.org/10.1007/978-3-030-01234-2>
- Cheng, H. Y., & Yu, C. C. (2015). Multi-model solar irradiance prediction based on automatic cloud classification. *Energy*, 91, 579–587. <https://doi.org/10.1016/j.energy.2015.08.075>
- Cote, M. P. (2022). *Cloud Image Classification Using Machine Learning*. Florida Institute of Technology.
- Guijo-Rubio, D., Gutiérrez, P. A., Casanova-Mateo, C., Fernández, J. C., Gómez-Orellana, A. M., Salvador-González, P., Salcedo-Sanz, S., & Hervás-Martínez, C. (2020). Prediction of convective clouds formation using evolutionary neural computation techniques. *Neural Computing and Applications*, 32(17), 13917–13929. <https://doi.org/10.1007/s00521-020-04795-w>
- Jin, W., Wang, L., Zeng, X., Liu, Z., & Fu, R. (2014). Classification of clouds in satellite imagery using over-complete dictionary via sparse representation. *Pattern Recognition Letters*, 49, 193–200. <https://doi.org/10.1016/j.patrec.2014.07.015>
- Juncklaus Martins, B., Marian Arrais, J., Perello Ricci Neto, G., Cerentini, A., Mantelli, S., & von Wangenheim, A. (2023). Semantic Segmentation of Cloud Images captured with Horizon-Oriented Cameras for Nowcasting Applications. *Research Square*. <https://doi.org/10.21203/rs.3.rs-3072958/v1>
- Lee, J., Weger, R. C., Sengupta, S. K., & Welch, R. M. (1990). A Neural Network Approach to Cloud Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 28(5), 846–855. <https://doi.org/10.1109/36.58972>
- Liu, Y., Xia, J., Shi, C. X., & Hong, Y. (2009). An improved cloud classification algorithm for China's FY-2C multi-channel images using artificial neural network. *Sensors*, 9(7), 5558–5579. <https://doi.org/10.3390/s90705558>
- Martus, C., & Johnson, B. (2024). Deep-learning-based cloud segmentation and classification for weather monitoring. *Remote Sensing of Clouds and the Atmosphere XXIX*. <https://doi.org/https://doi.org/10.1117/12.3031208>
- Nataraja, V., Schmidt, S., Chen, H., Yamaguchi, T., Kazil, J., Feingold, G., Wolf, K., & Iwabuchi, H. (2022). Segmentation-based multi-pixel cloud optical thickness retrieval using a convolutional neural network. *Atmospheric Measurement Techniques*, 15(17), 5181–5205. <https://doi.org/10.5194/amt-15-5181-2022>
- Noia, A. Di, & Hasekamp, O. P. (2018). Neural Networks and Support Vector Machines and Their Application to Aerosol and Cloud Remote Sensing: A Review. Kokhanovsky A. (Eds) *Springer Series in Light Scattering. Springer Series in Light Scattering*. https://doi.org/https://doi.org/10.1007/978-3-319-70796-9_4
- Nugraheny, D. (2015). Metode Nilai Jarak Guna Kesamaan Atau Kemiripan Ciri Suatu Citra (Kasus Deteksi Awan Cumulonimbus Menggunakan Principal Component Analysis). *Angkasa: Jurnal*

- Ilmiah Bidang Teknologi*, 7(2), 21. <https://doi.org/10.28989/angkasa.v7i2.145>
- Nurrani, H., Nugroho, A. K., & Heranurweni, S. (2023). Image Classification of Vegetable Quality using Support Vector Machine based on Convolutional Neural Network. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(1), 168–178. <https://doi.org/10.29207/resti.v7i1.4715>
- Omar, M. B., Ibrahim, R. Bin, Bingi, K., Sambaragi, S., Deshannavar, U., & Anawar, M. A. Bin. (2023). Convective Cloud Classification System using Convolution Neural Network. *IEEE International Conference on Computing (ICOCO)*. <https://doi.org/10.1109/ICOCO59262.2023.10397598>
- Purbantoro, B., Aminuddin, J., Manago, N., Toyoshima, K., Tetuko, J., Sumantyo, S., & Kuze, H. (2019). Evaluation of Cloud Type Classification based on Split Window Algorithm using Himawari-8 Satellite Data. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 170–173.
- Putra, Angie; Lursinsap, C. (2014). Cumulonimbues Prediction Using Artificial Neural Network Back Propagation with Radiosonde Indeces. *National Remote Sensing Seminar*, 96(24), 7549–7551.
- Putra, R. M., Fibriantika, E., Herawati, H., Kusumayanti, Y., Afriani, E., Hidayanti, A., Wiujiana, A., Swastiko, W. A., & Andrianto, D. (2021). Cumulonimbus cloud prediction based on machine learning approach using radiosonde data in Surabaya, Indonesia. *IOP Conference Series: Earth and Environmental Science*, 724(1). <https://doi.org/10.1088/1755-1315/724/1/012047>
- Rais, A. F., Putra, R. M., Fitrianto, M. A., & Hermansyah, T. (2022). A Preliminary Comparative Study on the Feasibility of a Multipurpose Numerical Weather Model for Prediction of Cumulonimbus Clouds in Indonesia. *International Conference on Science and Technology (ICOSTECH)*. <https://doi.org/10.1109/ICOSTECH54296.2022.9829049>
- Rajendiran, N., Sebastian, S., & Kumar, L. S. (2024). Cloud Segmentation, Validation of Weather Data, and Precipitation Prediction Using Machine Learning Algorithms. *Arabian Journal for Science and Engineering*, 49, 12259–12289. <https://link.springer.com/article/10.1007/s13369-023-08611-0>
- Rozi, M. F. (2019). *Prediksi Pertumbuhan Awan Cumulonimbus pada Citra Himawari IR Enhanced Menggunakan Deep Echo State Network (DEEPESN)*. Universitas Islam Negeri Sunan Ampel.
- Rudi, A. (2014). 2014, Rudi, Hilangnya pesawat. *Compas.Com*. <https://nasional.kompas.com/read/2014/12/29/18163231/BMKG.Ungkap.Alasan.Awan.Kumulonimbus.Bahaya.untuk.Penerbangan>
- Sebastian, S., Kumar, L. S., & Annadurai, P. (2021). Segmentation of satellite images using machine learning algorithms for cloud classification. *Indian Journal of Radio and Space Physics*, 50(1), 12–18.
- Segal-Rozenhaimer, M., Li, A., Das, K., & Chirayath, V. (2020). Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN). *Remote Sensing of Environment*, 237. <https://doi.org/10.1016/j.rse.2019.111446>
- Shen, X., Tian, X., Liu, T., Xu, F., & Tao, D. (2018). Continuous Dropout. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9), 3926–3937. <https://doi.org/10.1109/TNNLS.2017.2750679>
- Shi, G., & Zuo, B. (2022). CloudRCNN: A Framework Based on Deep Neural Networks for Semantic Segmentation of Satellite Cloud Images. *Applied Sciences (Switzerland)*, 12(11). <https://doi.org/10.3390/app12115370>
- Sinko, M., Kamencay, P., Sykora, P., Benco, M., & Hudec, R. (2019). Cloud-type classification of ground-based images using deep learning. *ICETA 2019 - 17th IEEE International Conference on Emerging ELearning Technologies and Applications, Proceedings*, 715–720. <https://doi.org/10.1109/ICETA48886.2019.9040149>
- Tapakis, R., & Charalambides, A. G. (2012). Equipment and methodologies for cloud detection and classification: A review. *Solar Energy*, 95, 392–430. <https://doi.org/10.1016/j.solener.2012.11.015>
- Uddstrom, M. J., & Gray, W. R. (1996). Satellite cloud classification and rain-rate estimation using multispectral radiances and measures of spatial texture. In *Journal of Applied Meteorology* (Vol. 35, Issue 6, pp. 839–858). [https://doi.org/10.1175/1520-0450\(1996\)035<0839:SCCARR>2.0.CO;2](https://doi.org/10.1175/1520-0450(1996)035<0839:SCCARR>2.0.CO;2)
- Wanayumini. (2021). *Supervised Image Classification untuk Fenomena Chaos Angin Puting Beliung Berdasarkan Average Correlation Angle* [Universitas Sumatera Utara]. https://doi.org/10.1007/0-306-47647-9_9
- Wang, C., Platnick, S., Meyer, K., Zhang, Z., & Zhou, Y. (2020). A machine-learning-based cloud detection

- and thermodynamic-phase classification algorithm using passive spectral observations. *Atmospheric Measurement Techniques*, 13(5), 2257–2277. <https://doi.org/10.5194/amt-13-2257-2020>
- Wang, Q., & Fan, W. (2010). Cumulonimbus forecasting based on rough set and artificial immune algorithm. *Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010*, 6, 2856–2860. <https://doi.org/10.1109/ICNC.2010.5584020>
- Yuwana, R. S., Fauziah, F., Heryana, A., Krisnandi, D., Kusumo, R. B. S., & Pardede, H. F. (2020). Data Augmentation using Adversarial Networks for Tea Diseases Detection. *Jurnal Elektronika Dan Telekomunikasi*, 20(1), 29–35. <https://doi.org/10.14203/jet.v20.29-35>
- Zhang, J., Liu, P., Zhang, F., & Song, Q. (2018). CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network. *Geophysical Research Letters*, 45(16), 8665–8672. <https://doi.org/10.1029/2018GL077787>
- Zhang, S., Bao, Y., Zhou, P., Jiang, H., & Dai, L. (2014). Improving Deep Neural Network for LVCSR using Dropout and Shrinking Structure. *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 6899–6903.
- Zhang, Z., Ma, Y., Li, Y., Li, F. W. B., Shum, H. P. H., Yang, B., Guo, J., & Liang, X. (2020). Cumuliform cloud formation control using parameter-predicting convolutional neural network. *Graphical Models*, 111(July), 101083. <https://doi.org/10.1016/j.gmod.2020.101083>
- Zhao, M., Chang, C. H., Xie, W., Xie, Z., & Hu, J. (2020). Cloud Shape Classification System Based on Multi-Channel CNN and Improved FDM. *IEEE Access*, 8, 44111–44124. <https://doi.org/10.1109/ACCESS.2020.2978090>
- Zhu, T., Wei, L., & Guo, Y. (2021). Cloud classification of ground-based cloud images based on convolutional neural network. *Journal of Physics: Conference Series*, 2035(1). <https://doi.org/10.1088/1742-6596/2035/1/012020>