

ANALISIS PERBANDINGAN TINGKAT PERFORMANSI METODE DT, RF, NB, KNN, SVM, DAN CNN UNTUK KLASIFIKASI CITRA CUACA

Christopher Evan^{1*}, Theofilus Arifin², Jason Jabez Cuwendi³, Henri Jayanata Kusuma⁴

¹²³⁴Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur

*Penulis Korespondensi

Abstrak

Cuaca merupakan keadaan udara atmosfer pada kurun waktu dan tempat tertentu yang sifatnya tidak menentu dan berubah-ubah. Pada jurnal ini, kami menerapkan algoritma *Machine Learning* yaitu Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) dan algoritma *Deep Neural Network* yaitu Convolutional Neural Network (CNN) untuk mengklasifikasikan cuaca dari gambar yang diperoleh dari *Multi-class Weather Dataset (MWD)*. Tujuan dari penelitian ini adalah untuk membandingkan tingkat performa dari enam algoritma tersebut. Kami menggunakan K-Fold Cross Validation dengan $cv = 5$ untuk memvalidasi model kami. Sebelum melakukan prediksi, gambar akan melalui tahap *preprocessing* terlebih dahulu dengan *smoothing* dan *sharpening*. Dari hasil uji coba, kami memperoleh akurasi prediksi sebesar 68.84% dengan algoritma DT, 81.89% dengan algoritma RF, 70.91 dengan algoritma NB, 74.47% dengan algoritma KNN, sebesar 83.97% dengan menggunakan algoritma SVM, dan sebesar 99.87% dengan menggunakan CNN.

Kata Kunci: klasifikasi, cuaca, preprocessing, algoritma, machine learning, deep neural network.

COMPARATIVE PERFORMANCE ANALYSIS OF DT, RF, NB, KNN, SVM, AND CNN METHOD FOR WEATHER IMAGE CLASSIFICATION

Abstract

Weather defines the state of the atmosphere during a certain period of time and in a certain area that is uncertain and fluctuating. In this journal, we implement Machine Learning algorithms that is Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Deep Neural Network algorithm in the form of Convolutional Neural Network (CNN) to classify weather from images using the Multi-class Weather Dataset (MWD). The aim of this study is to compare performance between those six algorithms. We use K-Fold Cross Validation with $cv = 5$ to validate our model. Before predicting the images, we preprocess the images using smoothing and sharpening. From our testing, our model achieves 68.84% prediction accuracy using DT, 81.99% using RF, 70.91 using NB, 74.47% using KNN, 83.97% using SVM, and 99.87% using CNN.

Keywords: classification, weather, preprocessing, algorithm, machine learning, deep neural network

1. PENDAHULUAN

1.1. Latar Belakang

Prakiraan cuaca berarti memprakirakan cuaca yang akan terjadi dan melihat bagaimana cuaca berganti dari waktu ke waktu. Perubahan cuaca umumnya terjadi karena pergerakan atau transfer energi. Banyak faktor seperti suhu, curah hujan, tekanan, kelembaban, cahaya matahari, angin dan keadaan mendung perlu diperhatikan untuk memprakirakan cuaca. Mengidentifikasi tipe awan yang berhubungan dengan pola cuaca juga memungkinkan untuk dilakukan. Berbagai pola cuaca yang ada membantu memprediksi ramalan cuaca.

Di masa lampau, orang-orang umumnya memakai tekanan barometrik, keadaan cuaca saat ini dan keadaan langit untuk

memprediksi sedangkan sekarang sudah ada banyak model berbasis komputer yang mempertimbangkan faktor-faktor atmosferik untuk memprediksi cuaca. Tetapi metode ini tidak akurat karena sifat alam dari atmosfer yang kacau dan terus berubah. Bahkan memprediksi cuaca dalam jangka waktu yang panjang dapat menurunkan akurasi, oleh karena itu banyak model prakiraan cuaca di zaman sekarang yang hanya dipakai beberapa hari maksimal 10 hari. Akurasi dari prediksi akan terus menurun semakin lama dipakai.

1.2. Tinjauan Literatur

Weather Forecasting using satellite image processing and ANN

The interpretation of satellite weather imagery has generally required the experience of a well-trained meteorologist. However, it is not always possible, or feasible to have an expert meteorologist on hand when such an interpretation system would be quite desirable. Also, to take advantage of this available data in a reasonable and useful time increment, the system must be efficient and have low implementation cost. There are 3 main types of satellite image available - Visible, Infrared and Water Vapor. Visible images are obtained only during the day they are used to determine the thickness of the clouds. Infrared images are obtained using special infrared sensors. The major advantage of this type is that it can be obtained even during night. It can be used to measure the temperature of cloud tops. Water Vapor images indicate the moisture content and humidity. The brighter areas tend to have high chances of rainfall.

Penulis dari artikel ini menggunakan gambar dari satelit untuk memprediksi cuaca. Untuk memisahkan gambar, Penulis melakukan normalisasi gambar dengan cara memisah campuran *matrix* yang ada dari *input* dan akan digunakan untuk membuat segmentasi awan. Ditulis di artikel ini bahwa salah satu langkah penting dalam sistem interpretasi gambar satelit cuaca otomatis adalah segmentasi gambar. Dalam proses ini, fitur awan yang signifikan akan diambil dari gambar dan dipersiapkan untuk proses selanjutnya. Hasil segmentasi akan dihasilkan dengan cepat dan mempunyai akurasi yang tinggi untuk diintegrasikan ke dalam sistem interpretasi cuaca otomatis atau untuk estimasi *cloud cover*. Prediksi pada saat ini sudah menghasilkan hasil yang positif dan akurasi prediksi sudah cukup tinggi, tetapi *error* seperti fenomena cuaca akan selalu tetap ada. Walau begitu fenomena cuaca juga diperlukan untuk membuat model yang paling efisien dan memiliki prediksi yang paling akurat.

1.3. Pernyataan Masalah

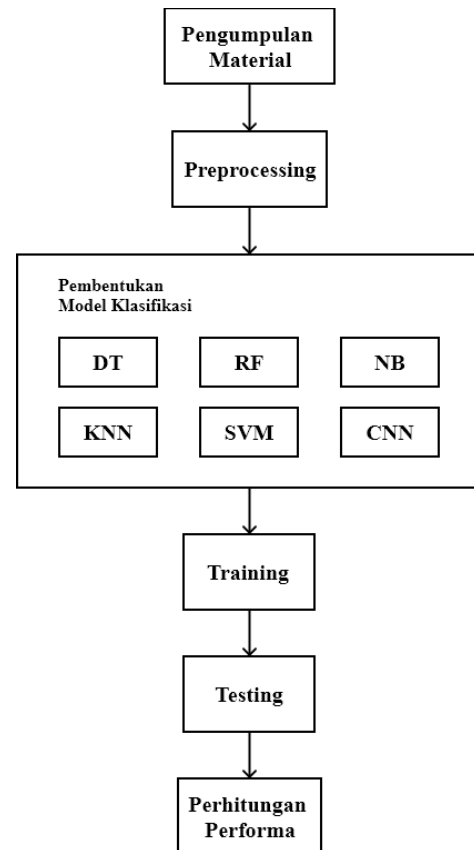
Akurasi dari banyak *classifier* yang ada seperti KNN, SVM, CNN, DT, RF, dan NB semakin berkurang semakin sering *classifier* itu dipakai. Adakalanya juga, *classifier* akan mengklasifikasikan gambar kurang tepat, karena gambar tersebut memiliki dua fitur bertentangan di dalamnya, seperti gambar hujan dengan awan di atasnya. Untuk dapat mengatasi masalah diatas, kami berusaha membuat proyek.

Proyek ini dibuat untuk mengklasifikasikan suatu gambar keadaan cuaca dan memberinya label. Sebelumnya, gambar akan di *preprocessing* dahulu untuk meningkatkan akurasi prediksi dari *classifier*. *Classifier* yang digunakan adalah KNN, SVM, CNN, DT, RF, dan NB. Label yang dimiliki *dataset* adalah "Rain", "Cloudy", "Shine", "Sunrise".

2. METODE PENELITIAN

Proses metodologi penelitian yang dilakukan dimulai dari pengumpulan material, pembentukan *classifier*, melakukan validasi menggunakan K-Fold Cross Validation, melakukan training pada *classifier*, dan melakukan testing.

Gambar 1. Alur Metode Penelitian

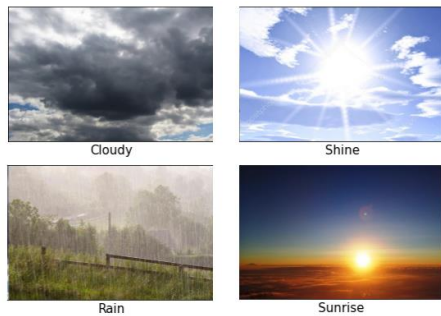


2.1. Pengumpulan Material

Material yang digunakan pada jurnal ini merupakan sebuah *dataset* yang berisikan kumpulan gambar dari 4 macam cuaca. *Dataset* yang digunakan adalah *Multi-class Weather Dataset* (MWD) yang di dalamnya meliputi cuaca *Cloudy* (Mendung), *Rain* (Hujan), *Shine* (Cerah), dan *Sunrise* (Matahari Terbit). *Dataset* yang digunakan memiliki total jumlah gambar sebanyak 1125 gambar dengan detail sebagai berikut:

Tabel 1. Detail *Dataset*

Label Cuaca	Jumlah Gambar	Persentase
Cloudy	300	26.7%
Rain	215	19.1%
Shine	253	22.5%
Sunrise	357	31.7%

Gambar 2. Contoh gambar cuaca pada *dataset*

2.2. Preprocessing

Sebelum digunakan untuk melakukan *training* maupun *testing* pada *classifier* yang akan diuji, dilakukan tahap *preprocessing* pada setiap gambar yang ada di *dataset*. Tahap *preprocessing* yang dilakukan meliputi:

Proses *smoothing* menggunakan *Gaussian Blur* dengan kernel 5x5, untuk menghilangkan *noise* yang ada pada gambar

Dilanjutkan dengan proses *sharpening* dengan menggunakan fungsi `filter2D()` yang terdapat pada *library* OpenCV untuk mengembalikan detail yang hilang setelah proses *smoothing*. Kernel yang digunakan adalah

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Proses selanjutnya adalah melakukan *resize* pada gambar menjadi resolusi 128x128.

Setelah dilakukan tahap *preprocessing*, kumpulan gambar pada *dataset* akan dipecah menggunakan fungsi `train_test_split()` yang terdapat pada *library* sklearn dengan detail *parameter* pada Tabel 2.

Tabel 2. Parameter *split*

Parameter	Deskripsi
random state	15

Proses *splitting* atau pemecahan ini bertujuan untuk mendapatkan kumpulan gambar acak dari *dataset* yang akan digunakan untuk *training* dan *testing*. Proses ini memecah *dataset* dengan presentase 70% dari *dataset* digunakan untuk *training* dan 30% dari *dataset* digunakan untuk *testing*.

Untuk menguji *classifier* menggunakan *single image* (gambar satuan), digunakan gambar cuaca (*cloudy/rain/shine/sunrise*) yang diambil secara acak dari internet.

2.3. Pembentukan Model Klasifikasi

Dalam jurnal ini, dipaparkan penggunaan 6 metode yang berbeda untuk melakukan klasifikasi gambar. Metode yang digunakan adalah DT(Decision Tree), RF (Random Forest), NB (Naïve Bayes), KNN (K-Nearest Neighbors), SVM (Support Vector Machine), CNN (Convolutional Neural Networks).

Pada algoritma DT (Decision Tree) parameter yang digunakan adalah maximum depth. Tabel 3 menunjukkan detail parameter DT yang digunakan.

Tabel 3. Parameter DT

Parameter	Deskripsi
maximum depth	5

Pada algoritma RF (Random Forest) parameter yang digunakan adalah jumlah estimator. Tabel 4 menunjukkan detail parameter RF yang digunakan.

Tabel 4. Parameter RF

Parameter	Deskripsi
jumlah estimator	100

Pada algoritma NB (Naïve Bayes) parameter yang digunakan adalah default parameter yang NB punya, yaitu prior probability lalu dilakukan smoothing dengan default float 1e-9 untuk menstabilkan kalkulasi.

Pada algoritma KNN (K-Nearest Neighbors) parameter yang digunakan adalah jumlah neighbors dan tipe distance. Tabel 5 menunjukkan detail parameter KNN yang digunakan pada project ini. Jumlah neighbors yang digunakan adalah 1, 5, 7 dan 9, sedangkan tipe distance yang digunakan adalah Manhattan, Euclidean, Minkowski dan Chebyshev.

Tabel 5. Parameter KNN

Parameter	Deskripsi
Jumlah Neighbors	1, 5, 7, 9
Tipe Distance	Manhattan, Euclidean, Minkowski, Chebyshev

Pada algoritma SVM (Support Vector Machine) parameter yang digunakan adalah jenis kernel. Tabel 6 menunjukkan detail parameter SVM yang digunakan.

Tabel 6. Parameter SVM

Parameter	Deskripsi
Kernel	RBF, Poly, Linear, Sigmoid

Pada algoritma CNN (Convolutional Neural Networks). Metode CNN yang digunakan menggunakan model *sequential* yang terdapat pada *library* tensorflow. Tabel 7 menunjukkan detail parameter CNN yang digunakan.

Tabel 7. Parameter CNN

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 126, 126, 126)	896
max_pooling2d_2 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_4 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_3 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_5 (Conv2D)	(None, 28, 28, 64)	36928
flatten_1 (Flatten)	(None, 50176)	0
dense_2 (Dense)	(None, 64)	3211328
dense_3 (Dense)	(None, 10)	650

2.4. Training

Data training diambil secara acak dari seluruh *image* yang ada. Seluruh *image* dibagi menjadi 30% *data training* dan 70% *data testing*. Setelah *data training* telah didapat dari pemilihan *random*, *preprocessing* akan dilakukan ke *data training*.

Training algoritma DT, RF, NB, KNN, dan SVM menggunakan kriteria *default* dari *library* *sklearn*, tanpa perubahan. Training algoritma CNN dilakukan dengan *library* TensorFlow, dengan kriteria *epoch* 25 dan adanya *callback* sebagai *output*.

Tabel 8 menunjukan nilai *cross validation* yang didapat untuk setiap algoritma *classifier*, kecuali CNN.

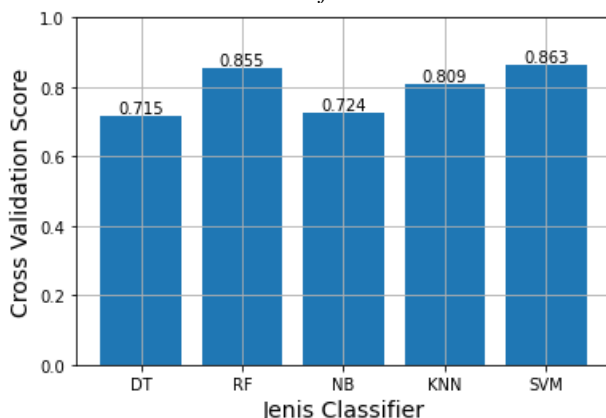
Tabel 8. Skor *Cross Validation*

Classifier	Cross Validation Score
DT	0.695
RF	0.845
NB	0.728
KNN	0.819
SVM	0.864

Gambar 3. Visualisasi *k-fold Cross Validation*



Gambar 4. Perbandingan skor *Cross Validation* setiap *classifier*



2.5. Testing

Testing akan dilakukan dengan algoritma *Stratified K-Fold Cross Validation* dengan banyak *K* 5. Ini menyebabkan testing menggunakan 20 persen dari *dataset* per-iterasi, dimana akan ada 5 iterasi.

Stratifikasi pada testing akan juga menyebabkan 20 persen dari *dataset* tadi diambil dari bagian-bagian

berbeda *dataset* (sehingga akan terbagi menjadi lebih dari satu bagian), dimana *cross validation non-stratified* akan menyebabkan 20 persen tadi akan diambil dalam satu bagian saja.

Testing dilakukan kepada seluruh algoritma *classifier*, kecuali CNN. Ini dikarenakan *cross validation* kepada algoritma CNN akan menyebabkan *runtime* dari program sangat panjang.

2.6. Perhitungan Performa

Naufal, 2021 menyatakan bahwa perhitungan Performa dilakukan dengan menghitung *accuracy*, *precision*, *recall*, dan *F1* dari masing-masing algoritma *classifier*. Rumus dari *accuracy*, *precision*, *recall*, dan *F1* dapat dilihat pada persamaan (1), (2), (3), dan (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 \text{ score} = \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Accuracy menunjukkan akurasi dari *classifier*. *Precision* menunjukkan rasio prediksi positif yang benar terhadap total prediksi positif. *Recall* menunjukkan rasio prediksi positif terhadap total prediksi positif. *F1* menunjukkan tingkat akurasi model *dataset*.

Tabel 9. Tabel performa tiap *classifier*

Classifier	Accuracy	Precision	Recall	F1
DT	0.91	0.9	0.95	0.46
RF	0.98	0.99	0.99	0.49
NB	0.89	0.94	0.88	0.46
KNN	0.98	0.99	0.99	0.49
SVM	0.98	1.0	0.97	0.49

3. HASIL DAN PEMBAHASAN

Dari hasil uji coba, kami memperoleh akurasi prediksi sebesar 68.84% dengan algoritma DT, 81.89% dengan algoritma RF, 70.91 dengan algoritma NB, 74.47% dengan algoritma KNN, sebesar 83.97% dengan menggunakan algoritma SVM, dan sebesar 99.87% dengan menggunakan CNN

Setelah dilakukan penjalanan terhadap program, terlihat bahwa keakuratan prediksi metode-metode sebelumnya berkisar antara 64.84% hingga 99.87%. Tabel 10 mengandung nilai-nilai keakuratan algoritma *classification* pada *dataset*.

Tabel 10. Keakuratan Algoritma Klasifikasi

Classifier	Akurasi
DT	64.84%
RF	81.89%
NB	70.91%
KNN	74.47%
SVM	83.97%
CNN	99.87%

Hasil akurasi ini menunjukkan bahwa CNN jelas merupakan algoritma *classifier* yang paling baik untuk mengklasifikasi *dataset* ini dalam kelas-kelas cuaca yang telah ditentukan.

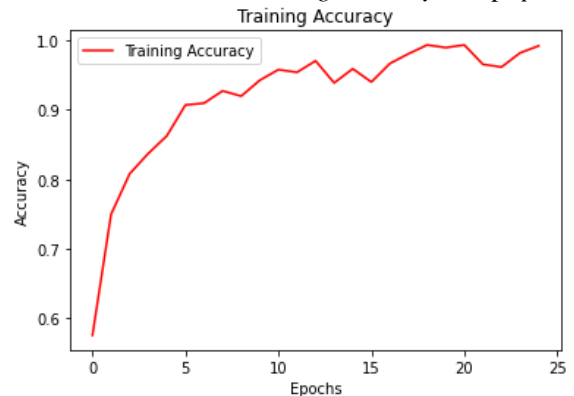
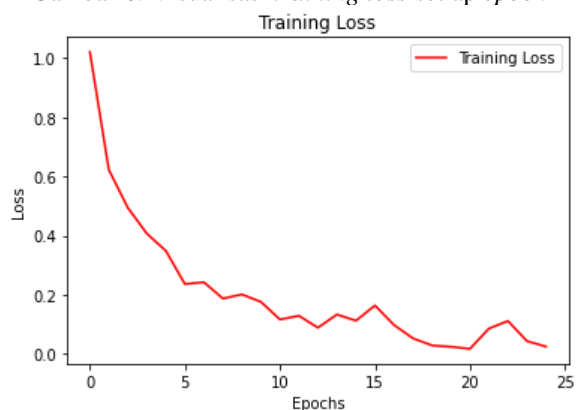
Kami juga melakukan pencatatan *runtime* (waktu berjalan) dari algoritma-algoritma tersebut, yang tersedia pada Tabel 11.

Tabel 11. Lama *Runtime* Algoritma *Classification*

Classifier	Runtime
DT	36.2s
RF	14.3s
NB	480 ms
KNN	8.7s
SVM	21.7s
CNN	5min 35s

Hasil ini berasal dari *library* Python time. Disini, terlihat bahwa CNN memakan waktu terlama, pada 5 menit 35 detik, dan NB memakan waktu tercepat, pada 0.48 detik. Dengan menghubungkan hasil lama *runtime* dengan hasil akurasi, dapat diinferensikan bahwa KNN merupakan algoritma yang cepat namun kurang akurat pada *dataset* ini, CNN merupakan algoritma yang sangat lambat namun sangat akurat pada *dataset* ini, dan SVM merupakan algoritma yang terdapat di pertengahan KNN dan CNN, dimana SVM memiliki keakuratan dan waktu *runtime* ditengah KNN dan CNN.

Perlu juga diperhatikan bahwa untuk *classifier* CNN, nilai-nilai yang tertera pada tabel-tabel sebelumnya didapatkan setelah CNN berjalan sebanyak 25 *epoch*. Peningkatan *epoch* akan meningkatkan *accuracy score* dan *runtime* per-*epoch*, berkisar dari *accuracy score* 63.2 persen pada epoch pertama, hingga 99.4 persen pada epoch ke dua puluh lima.

Gambar 5. Visualisasi *training accuracy* setiap *epoch*Gambar 6. Visualisasi *training loss* setiap *epoch*

4. KESIMPULAN

Dari hasil penelitian dan pembahasan yang sudah dilakukan dapat ditarik kesimpulan bahwa algoritma CNN memiliki tingkat performa yang paling tinggi dibandingkan dengan algoritma lain dalam melakukan klasifikasi citra cuaca dari gambar yang diperoleh dari *Multi-class Weather Dataset (MWD)*. Algoritma CNN mencapai tingkat performa hingga 99.87% dengan waktu *runtime* selama 5 menit 35 detik.

REFERENSI

- [1] Kapadia, N. S., Rana, D. P., & Parikh, U. (2016). Weather Forecasting using Satellite Image Processing and Artificial Neural Networks. *International Journal of Computer Science and Information Security*, 14(11), 1069.
- [2] Naufal, M. F. (2021). Analisis Perbandingan Algoritma Svm, Knn, Dan Cnn untuk Klasifikasi Citra Cuaca. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(2), 311-317.
- [3] Singh, N., Chaturvedi, S., & Akhter, S. (2019). Weather forecasting using machine learning algorithm. *2019 International Conference on Signal Processing and Communication (ICSC)*, 171-174.
- [4] Guerra, J. C. V., Khanam, Z., Ehsan, S., Stolkin, R., & McDonald-Maier, K. (2018). Weather Classification: A new multi-class dataset, data augmentation approach and

comprehensive evaluations of Convolutional Neural Networks. *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 305–310.

- [5] Purwandari, K., Sigalingging, J. W. C., Cenggoro, T. W., & Pardamean, B. (2021). Multi-class Weather Forecasting from Twitter Using Machine Learning Approaches. *Procedia Computer Science*, 179, 47–54.
-