

## ANALISIS PERBANDINGAN ALGORITMA SVM, KNN, DAN CNN UNTUK KLASIFIKASI CITRA CUACA

Mohammad Farid Naufal<sup>\*1</sup>

<sup>1</sup>Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur  
Email: <sup>1</sup>\*faridnaufal@staff.ubaya.ac.id,  
<sup>\*</sup>Penulis Korespondensi

(Naskah masuk: 31 Desember 2020, diterima untuk diterbitkan: 22 Maret 2021)

### Abstrak

Cuaca merupakan faktor penting yang dipertimbangkan untuk berbagai pengambilan keputusan. Klasifikasi cuaca manual oleh manusia membutuhkan waktu yang lama dan inkonsistensi. *Computer vision* adalah cabang ilmu yang digunakan komputer untuk mengenali atau melakukan klasifikasi citra. Hal ini dapat membantu pengembangan *self autonomous machine* agar tidak bergantung pada koneksi internet dan dapat melakukan kalkulasi sendiri secara *real time*. Terdapat beberapa algoritma klasifikasi citra populer yaitu K-Nearest Neighbors (KNN), Support Vector Machine (SVM), dan Convolutional Neural Network (CNN). KNN dan SVM merupakan algoritma klasifikasi dari *Machine Learning* sedangkan CNN merupakan algoritma klasifikasi dari Deep Neural Network. Penelitian ini bertujuan untuk membandingkan performa dari tiga algoritma tersebut sehingga diketahui berapa gap performa diantara ketiganya. Arsitektur uji coba yang dilakukan adalah menggunakan 5 cross validation. Beberapa parameter digunakan untuk mengkonfigurasi algoritma KNN, SVM, dan CNN. Dari hasil uji coba yang dilakukan CNN memiliki performa terbaik dengan akurasi 0.942, precision 0.943, recall 0.942, dan F1 Score 0.942.

**Kata kunci:** klasifikasi cuaca, KNN, SVM, CNN, machine learning, deep neural network

## COMPARATIVE ANALYSIS OF IMAGE CLASSIFICATION ALGORITHM FOR WEATHER DATASET

### Abstract

Weather is an important factor that is considered for various decision making. Manual weather classification by humans is time consuming and inconsistent. Computer vision is a branch of science that computers use to recognize or classify images. This can help develop self-autonomous machines so that they are not dependent on an internet connection and can perform their own calculations in real time. There are several popular image classification algorithms, namely K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Convolutional Neural Network (CNN). KNN and SVM are Machine Learning classification algorithms, while CNN is a Deep Neural Networks classification algorithm. This study aims to compare the performance of that three algorithms so that the performance gap between the three is known. The test architecture is using 5 cross validation. Several parameters are used to configure the KNN, SVM, and CNN algorithms. From the test results conducted by CNN, it has the best performance with 0.942 accuracy, 0.943 precision, 0.942 recall, and F1 Score 0.942.

**Keywords:** weather classification, KNN, SVM, CNN, machine learning, deep neural network

### 1. PENDAHULUAN

Dengan berkembangnya teknologi, komputer memiliki kemampuan untuk melakukan berbagai macam hal. Komputer dapat meramal cuaca dengan melakukan pengamatan terhadap gambar satelit dan menentukan cuaca pada hari tersebut dan melakukan ramalan untuk cuaca selanjutnya (Automotive Revolution & Perspective Towards 2030, 2016).

Dengan koneksi internet semua komputer dapat mengakses informasi tersebut. Namun cuaca merupakan informasi yang memiliki perbedaan antara satu tempat dengan tempat lainnya. Walaupun komputer mendapatkan informasi cuaca dari internet belum tentu data tersebut sama dengan lokasi komputer tersebut. Sebagai contoh self-driving car dapat menggunakan data cuaca untuk mengatur

kecepatan dan mengaktifkan wiper. Data cuaca juga dapat digunakan untuk menginformasikan mobil dalam mengambil keputusan real time.

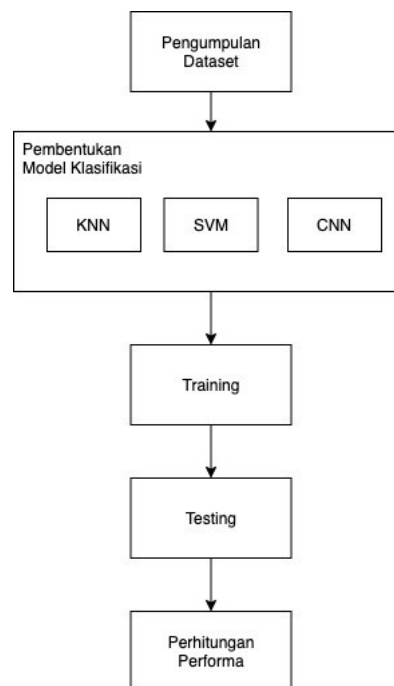
Image classification merupakan salah satu bidang yang diminati karena mampu menggantikan kemampuan visual manusia (Javidi, 2002). Dengan menggunakan image classification maka komputer dapat mengetahui cuaca hanya berdasarkan gambar secara real time. Dengan aplikasi image classification dapat membantu dalam pengembangan self-autonomous machine atau Advance Driver Assistance System (ADAS) (Kang, Chou and Fu, 2019).

(An, Chen and Shin, 2019) mengaplikasikan citra cuaca dengan menggunakan algoritma CNN untuk melakukan feature extraction dan mengkombinasikan dengan Multi-Class SVM. (Kang, Chou and Fu, 2019) melakukan klasifikasi cuaca menjadi 3 kelas yaitu hazy, rainy, dan snowy menggunakan CNN. (Xia et al., 2020) melakukan perbandingan terhadap beberapa arsitektur CNN yaitu AlexNet, VGG, dan GoogleNet untuk klasifikasi citra cuaca menjadi 4 kelas yaitu foggy, rainy, snowy, dan sunny. (Elhoseiny, Huang and Elgammal, 2015) melakukan klasifikasi citra cuaca menggunakan CNN dengan arsitektur AlexNet. (Ibrahim, Haworth and Cheng, 2019) menggunakan CNN dengan arsitektur ResNet50 untuk klasifikasi cuaca menjadi 3 kelas, yaitu rainy, snowy, dan foggy. Dari beberapa penelitian tersebut belum ada perbandingan algoritma klasifikasi Deep Neural Network dengan algoritma klasik Machine Learning seperti SVM dan KNN untuk melihat berapa gap performa diantara keduanya. Selain itu waktu eksekusi untuk proses training dan testing juga perlu dianalisis untuk melihat seberapa efektif *time to performance* diantara ketiga algoritma tersebut. Penelitian ini mencoba membandingkan algoritma CNN dengan algoritma Machine Learning klasik yaitu KNN dan SVM untuk melihat perbandingan performa dan waktu eksekusi di antara ketiga algoritma tersebut.

Sistematika penulisan pada penelitian ini terdiri dari 4 bagian. Pada bab 1 dijelaskan mengenai pendahuluan dan latar belakang permasalahan, pada bab 2 dijelaskan mengenai metode penelitian, pada bab 3 dijelaskan mengenai hasil dan pembahasan, dan bagian akhir yaitu bab 4 dijelaskan mengenai kesimpulan penelitian.

## 2. METODE PENELITIAN

Metodologi penelitian yang dilakukan terdiri dari pengumpulan dataset, pembentukan model klasifikasi, training model klasifikasi, testing, dan perhitungan performa. Gambar 1 menunjukkan alur metodologi penelitian.



Gambar 1. Alur metodologi penelitian

### 2.1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini adalah citra cuaca yang didapatkan dari *Multi-class Weather dataset for image classification* pada repositori *Mendeley* (Ajayi, 2018). Terdapat empat kelas cuaca, yaitu *cloudy*, *rain*, *shine*, dan *sunrise*. Tabel 1 menunjukkan jumlah dataset dari setiap kelas cuaca. Gambar 2 menunjukkan contoh dataset dari masing-masing kelas. Dataset yang digunakan diresize menjadi ukuran 64x64 sebelum dilakukan training dan testing

Tabel 1. Detail jumlah dataset dari setiap kelas cuaca

Kelas	Jumlah
Cloudy	298
Rain	214
Shine	251
Sunrise	357



Gambar 2. Contoh dataset cuaca

## 2.2. Pembentukan Model Klasifikasi

Pada tahapan ini setiap algoritma dikonfigurasi menggunakan beberapa parameter. Tujuannya adalah untuk mengetahui pengaruh parameter terhadap performa yang dihasilkan.

Pada algoritma KNN parameter yang diujicobakan adalah tipe distance dan jumlah neighbors. Tabel 2 menunjukkan parameter dari KNN yang digunakan dalam penelitian ini. Jumlah neighbors yang digunakan adalah 5, 7, dan 9, sedangkan tipe distance yang digunakan adalah Euclidean (Dokmanic et al., 2015) dan Minkowski (Çolakoglu, 2019).

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

Pada algoritma SVM parameter yang diujicobakan adalah jenis kernel. Tabel 3 menunjukkan parameter dari SVM.

Parameter	Deskripsi
Kernel	Linear, Poly, RBF

Pada algoritma CNN parameter yang diujicobakan adalah jumlah epoch, tipe convolution, tipe activation function, dan jumlah dense layer. Tabel 4 menunjukkan parameter dari CNN. Arsitektur CNN yang digunakan adalah LeNet (Lecun et al., 1998)

Layer (type)	Output Shape	Deskripsi
conv2d (Conv2D)	(None, 62, 62, 32)	Filter_size = 3x3. Act = ReLu
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	Pool_size = 2
conv2d_1 (Conv2D)	(None, 29, 29, 32)	Filter_size = 3x3. Act = ReLu
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	Pool_size = 2
flatten (Flatten)	(None, 6272)	-
dense (Dense)	(None, 128)	Act = ReLu
dense_1 (Dense)	(None, 4)	Act= Softmax
epoch	-	50
Optimizer	-	Adam
Batch size	-	8

Activation function yang digunakan pada proses convolution dan dense layer pertama adalah Rectified Linear Unit (ReLU) (Agarap, 2018). Persamaan (1) menunjukkan rumus activation function dari ReLu,  $z$  adalah nilai input activation function.

$$R(z) = \max(0, z) \quad (1)$$

Pada dense layer output activation function yang digunakan adalah Softmax (Zeiler and Fergus, 2014).

Hal ini dikarenakan kelas klasifikasi yang dihasilkan berjumlah empat. Persamaan (2) menunjukkan rumus activation function Softmax dengan input  $x_i$ , jumlah label class  $n$ , dan label kelas ke- $j$ .

$$S(x_i) = \frac{e^{x_i}}{\sum_j^n e^{x_j}} \quad (2)$$

Model klasifikasi yang dibentuk dalam penelitian ini dijalankan di sebuah perangkat keras komputer cloud dari Google Colaboratory (Google Colab, 2020). Detail spesifikasi dari perangkat keras komputer yang digunakan dapat dilihat pada Tabel 5.

Parameter	Spesifikasi
CPU	Intel® Xeon®, 2.30 GHz, 2 cores
RAM	12 GB
Space of Disk	25 GB
GPU Model Name	Nvidia K80, 12 GB

## 2.3. Training

Pada tahapan ini dilakukan training pada model klasifikasi yang telah dibuat. Training dilakukan menggunakan 80% dari dataset secara acak. Training diulang dan dilakukan sebanyak 5 kali. Dari tiap kali training dilakukan uji coba menggunakan cross validation. Detail dari penggunaan cross validation akan dijelaskan lebih detail pada tahapan testing.

Training algoritma KNN dilakukan menggunakan parameter yang dijelaskan pada Tabel 2. Begitu pula untuk algoritma SVM dan CNN training dilakukan masing-masing menggunakan parameter seperti pada Tabel 3 dan 4.

Model CNN yang digunakan dalam proses training ini juga menggunakan tahapan data augmentation dengan tujuan untuk menghindari overfitting. Data augmentation menggunakan library yang disediakan oleh keras (Chollet and &, 2020). Data augmentation yang digunakan terdiri dari horizontal flip, shear range dengan nilai 0.2 dan zoom range dengan nilai 0.2.

Horizontal flip digunakan untuk membuat gambar lebih bervariasi karena data training ditambahkan dengan gambar yang dirotasi secara horizontal 90 derajat. Shear range menggunakan shear transformation (Goldman, 1991) untuk membuat gambar lebih bervariasi dengan derajat rotasi tertentu, dan zoom range digunakan untuk memperbesar gambar dengan persentasi tertentu terhadap gambar asli.

## 2.4. Testing

Testing dilakukan menggunakan 20% dari dataset cuaca secara acak. Testing dilakukan dan diulang sebanyak 5 kali. Testing dilakukan menggunakan cross validation.

Pada tiap algoritma dilihat performa dari setiap cross validation. Hal ini dilakukan untuk melihat apakah performa yang dihasilkan stabil atau tidak.

Pada algoritma CNN di setiap epoch dilakukan proses validasi dengan data testing untuk dilihat akurasi. Namun akurasi yang didapatkan dari proses validasi data testing tidak digunakan untuk memperbarui bobot pada layer. Sedangkan akurasi dari model yang didapatkan dari validasi data training digunakan untuk memperbarui bobot pada layer. Optimasi yang digunakan adalah Adam Optimizer (Kingma and Ba, 2015).

## 2.5. Perhitungan Performa

Pada tahapan ini dilakukan perhitungan performa dari algoritma KNN, SVM, dan CNN. Performa yang dihitung adalah akurasi, precision, recall, dan F1 Score. Persamaan (3) (4) (5) (6) berturut turut menunjukkan rumus perhitungan akurasi, precision, recall, dan F1 Score. TP adalah True Positive, TN adalah True Negative, FP adalah False Positive, dan FN adalah False Negative.

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

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

Studi kasus pada penelitian ini adalah klasifikasi multiclass, sehingga untuk menghitung performa precision, recall, dan F1 Score menggunakan weighted metric (Goutte and Gaussier, 2005). Hal ini dilakukan dikarenakan jumlah dataset dari tiap kelas berbeda. Persamaan (7) menunjukkan rumus perhitungan weighted metric,  $m_i$  adalah metric precision, recall, atau F1 Score untuk class  $i$ ,  $j$  adalah jumlah class, dan  $c_i$  adalah jumlah data dari kelas  $i$ .

$$W_m = \frac{\sum_i^j m_i c_i}{\sum_i^j c_i} \quad (7)$$

## 3. HASIL DAN PEMBAHASAN

### 3.1. KNN

Tabel 6 menunjukkan hasil performa dari algoritma KNN. NN adalah Number Neighbors (Jumlah Neighbors), CV adalah Cross Validation, AVG adalah rata-rata performa dari tiap CV, dan AVG Perf adalah penjumlahan dari accuracy, precision, recall, dan f1 score dibagi dengan 4. AVG Perf untuk melihat rata-rata metric performa yang dihasilkan oleh tiap algoritma.

Tabel 6. Performa KNN

Dist	NN	Metric	CV1	CV2	CV3	CV4	CV5	AVG	AVG PERF
EUCLIDEAN	5	acc	0.746	0.754	0.723	0.777	0.768	0.754	0.766
		prec	0.794	0.783	0.769	0.806	0.802	0.791	
		rec	0.746	0.754	0.723	0.777	0.768	0.754	
		f1 score	0.755	0.761	0.731	0.783	0.774	0.761	
	7	acc	0.723	0.746	0.719	0.763	0.772	0.745	0.757
		prec	0.775	0.769	0.768	0.802	0.801	0.783	
		rec	0.723	0.746	0.719	0.763	0.772	0.745	
		f1 score	0.732	0.751	0.728	0.770	0.774	0.751	
	9	acc	0.723	0.746	0.719	0.763	0.772	0.745	0.760
		prec	0.794	0.783	0.769	0.806	0.802	0.791	
		rec	0.723	0.746	0.719	0.763	0.772	0.745	
		f1 score	0.732	0.751	0.728	0.770	0.774	0.751	
MINKOWSKI	5	acc	0.746	0.754	0.723	0.777	0.768	0.754	0.765
		prec	0.794	0.783	0.769	0.806	0.802	0.791	
		rec	0.746	0.754	0.723	0.754	0.768	0.749	
		f1 score	0.755	0.761	0.731	0.761	0.774	0.756	
	7	acc	0.723	0.746	0.719	0.763	0.772	0.745	0.756
		prec	0.775	0.769	0.768	0.802	0.801	0.783	
		rec	0.723	0.746	0.719	0.746	0.772	0.741	
		f1 score	0.732	0.751	0.728	0.751	0.774	0.747	
	9	acc	0.723	0.746	0.719	0.763	0.772	0.745	0.759
		prec	0.794	0.783	0.769	0.806	0.802	0.791	
		rec	0.723	0.746	0.719	0.746	0.772	0.741	
		f1 score	0.732	0.751	0.728	0.751	0.774	0.747	

Berdasarkan perhitungan performa yang dilakukan, algoritma KNN dengan distance Euclidean dan jumlah neighbors 5 memiliki performa terbaik. Perbedaan performa memang tidak signifikan, namun KNN dengan distance Euclidean dan jumlah neighbors 5 akan dibandingkan performanya dengan SVM dan CNN.

### 3.2 SVM

Tabel 7 menunjukkan performa algoritma SVM. SVM dengan kernel RBF memiliki performa terbaik jika dibandingkan dengan kernel lain. Semua metric performa dari kernel RBF unggul dari kernel yang lain. SVM dengan kernel RBF akan dibandingkan performanya dengan KNN dan CNN.

Tabel 7. Performa SVM

Kernel	Metric	CV1	CV2	CV3	CV4	CV5	AVG	AVG PERF
LINEAR	acc	0.857	0.871	0.804	0.862	0.804	0.839	0.845
	prec	0.858	0.876	0.809	0.867	0.867	0.855	
	rec	0.857	0.871	0.804	0.862	0.804	0.839	
	f1 score	0.858	0.870	0.805	0.864	0.806	0.841	
POLY	acc	0.862	0.875	0.795	0.875	0.813	0.844	0.849
	prec	0.863	0.877	0.799	0.877	0.877	0.859	
	rec	0.862	0.875	0.795	0.875	0.813	0.844	
	f1 score	0.862	0.876	0.797	0.875	0.814	0.845	
RBF	acc	0.875	0.848	0.813	0.888	0.862	0.857	0.860
	prec	0.876	0.850	0.823	0.889	0.889	0.865	
	rec	0.875	0.848	0.813	0.888	0.862	0.857	
	f1 score	0.875	0.848	0.815	0.888	0.862	0.858	

### 3.3 CNN

Tabel 8 menunjukkan hasil performa dari algoritma CNN. Dapat dilihat bahwa CNN memiliki performa yang sangat baik, semua metric performa yang dihasilkan memiliki nilai di atas 0.92.

Tabel 8. Performa CNN

Metric	CV1	CV2	CV3	CV4	CV5	AVG	AVG PERF
acc	0.946	0.951	0.942	0.942	0.929	0.942	0.942
prec	0.947	0.953	0.946	0.943	0.928	0.943	
rec	0.946	0.951	0.942	0.942	0.929	0.942	
f1score	0.946	0.951	0.942	0.942	0.928	0.942	

### 3.4 Perbandingan Performa

Tabel 9 menunjukkan perbandingan performa antara algoritma KNN, SVM, dan CNN. Untuk algoritma KNN, penelitian ini memilih KNN dengan distance Euclidean dan jumlah neighbors 5 dikarenakan memiliki performa terbaik dibandingkan KNN dengan parameter lainnya. Sedangkan untuk algoritma SVM, penelitian ini memilih SVM dengan kernel RBF karena memiliki performa terbaik diantara SVM dengan parameter lainnya. Gambar 3 menunjukkan grafik perbandingan performa antara algoritma KNN, SVM, dan CNN.

Tabel 9. Perbandingan Performa

Algoritma	Metric	CV1	CV2	CV3	CV4	CV5	AVG	AVG PERF
KNN 5 Euclidean	acc	0.746	0.754	0.723	0.777	0.768	0.754	0.766
	prec	0.794	0.783	0.769	0.806	0.802	0.791	
	rec	0.746	0.754	0.723	0.777	0.768	0.754	
	f1score	0.755	0.761	0.731	0.783	0.774	0.761	
SVM RBF	acc	0.875	0.848	0.813	0.888	0.862	0.857	0.860
	prec	0.876	0.850	0.823	0.889	0.889	0.865	
	rec	0.875	0.848	0.813	0.888	0.862	0.857	
	f1score	0.875	0.848	0.815	0.888	0.862	0.858	
CNN	acc	0.946	0.951	0.942	0.942	0.929	0.942	0.942
	prec	0.947	0.953	0.946	0.943	0.928	0.943	
	rec	0.946	0.951	0.942	0.942	0.929	0.942	
	f1score	0.946	0.951	0.942	0.942	0.928	0.942	

Penelitian ini juga membandingkan waktu eksekusi training dan testing dari tiap algoritma. Tabel 10 menunjukkan waktu eksekusi di setiap cross validation dalam satuan detik dari algoritma KNN dengan distance Euclidean dan jumlah neighbor 5, SVM dengan kernel RBF, dan CNN dengan jumlah epoch 50.

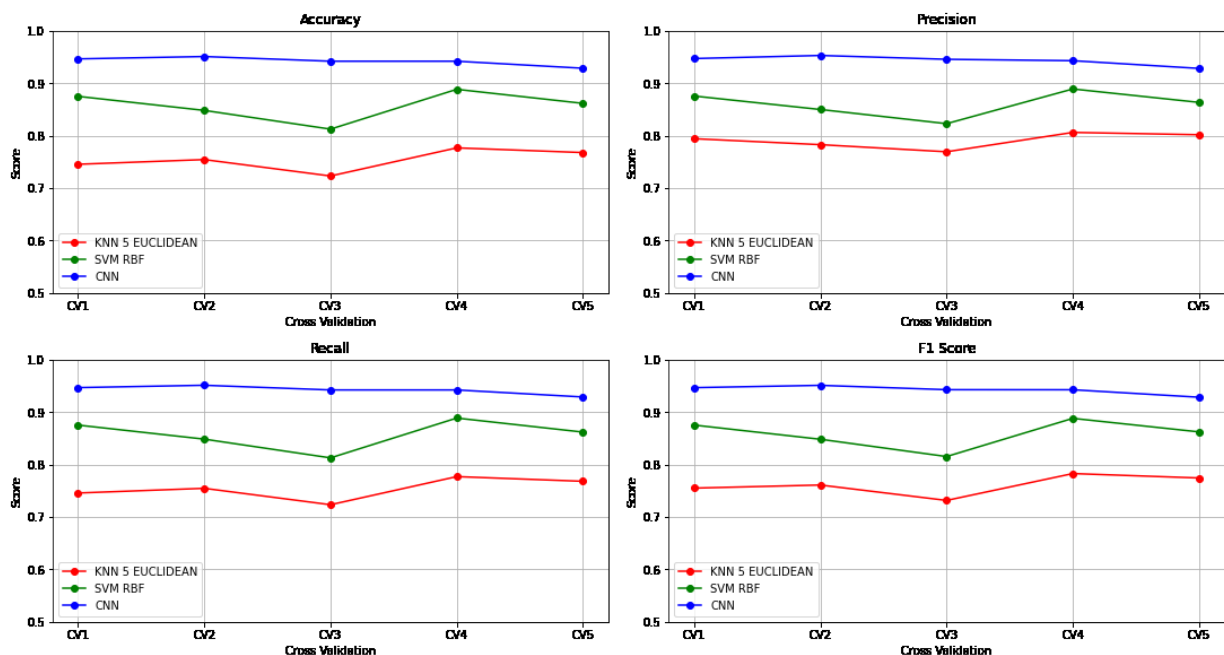
Tabel 10. Waktu eksekusi training dan testing (detik)

Algoritma	CV1	CV2	CV3	CV4	CV5	AVG
KNN	5.33	5.35	5.32	5.34	5.32	5.33
SVM	10.54	10.58	10.22	10.82	10.52	10.54
CNN	458.49	451.17	457.63	461.54	463.62	458.49

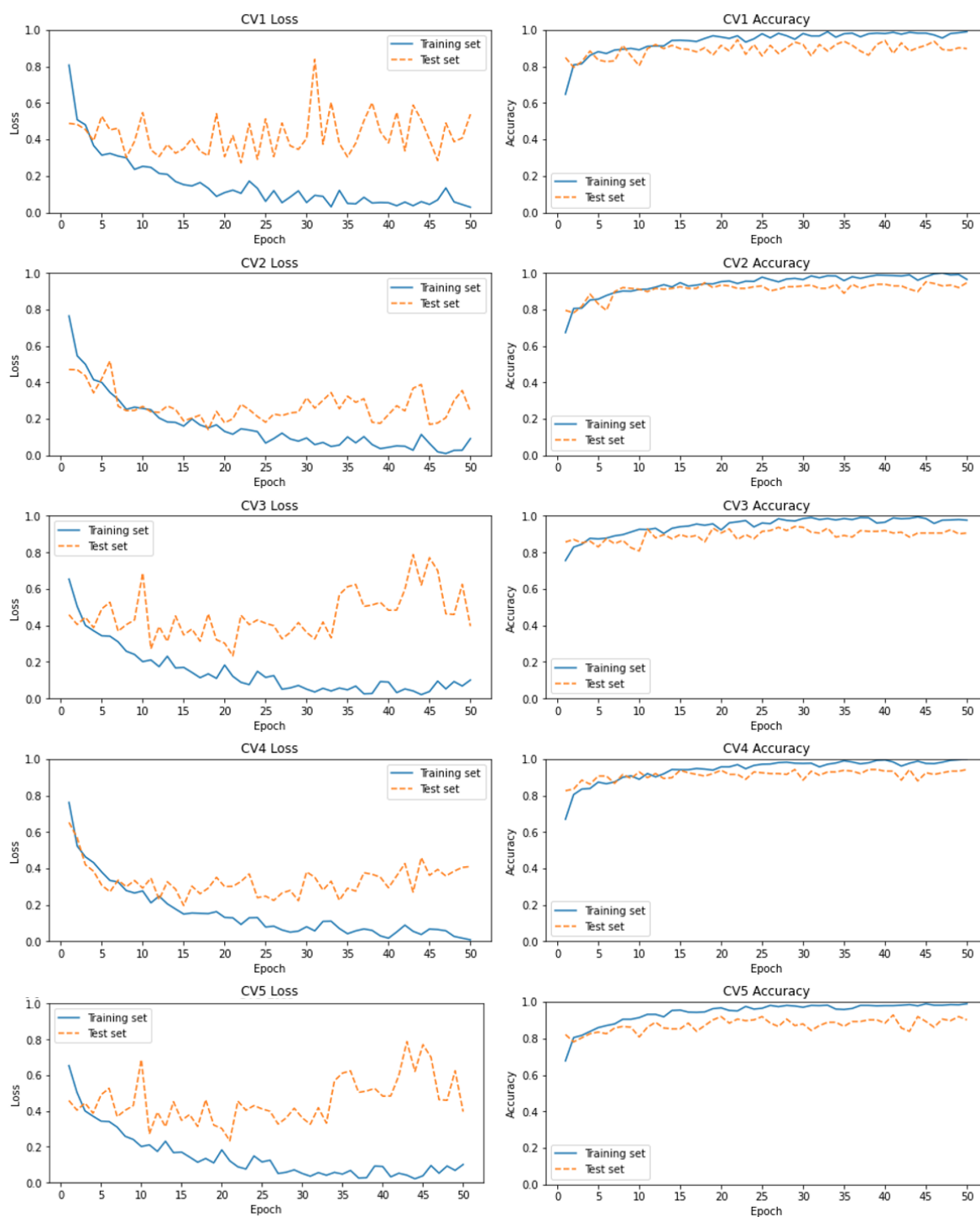
Berdasarkan Tabel 10 terlihat bahwa algoritma CNN memiliki waktu eksekusi paling lama jika dibandingkan dengan KNN dan SVM. Hal ini bergantung dari jumlah epoch yang digunakan.

Pada uji coba CNN, penelitian ini menggunakan epoch sebanyak 50. Berdasarkan uji coba rata-rata waktu eksekusi di tiap epoch adalah 9.82 detik dan menghasilkan rata-rata akurasi data testing mencapai 0.88 di epoch pertama dengan batch size 8. Performa akurasi ini cukup meyakinkan jika dibandingkan dengan algoritma SVM yang membutuhkan waktu rata-rata 10.54 detik untuk mendapatkan rata-rata akurasi 0.857. Akurasi CNN akan terus membaik hingga epoch ke-50 dan mendapatkan rata-rata akurasi sebesar 0.942.

CNN mendapatkan akurasi yang paling baik dibandingkan dengan KNN dan SVM. Walaupun waktu eksekusi CNN untuk mendapatkan akurasi maksimal membutuhkan waktu rata-rata 458.49 detik, performa yang dihasilkan cukup signifikan jika dibandingkan dengan KNN dan SVM.



Gambar 3. Perbandingan performa algoritma KNN, SVM, dan CNN di tiap Cross Validation



Gambar 4. Grafik Loss dan Accuracy CNN

Gambar 4 menunjukkan grafik loss dan accuracy dari algoritma CNN. Dapat dilihat bahwa metric loss dan accuracy dari data training semakin membaik di setiap epoch. Accuracy yang membaik di setiap epoch untuk data testing menandakan bahwa model yang dibuat tidak overfitting.

#### 4. KESIMPULAN

Berdasarkan uji coba yang dilakukan penelitian ini, algoritma CNN memiliki performa terbaik dalam

melakukan klasifikasi pada dataset cuaca yang diperoleh dari *Multi-class Weather dataset for image classification* pada repositori Mendeley (Ajayi, 2018). Performa yang didapatkan oleh CNN adalah accuracy sebesar 0.942, precision sebesar 0.943, recall sebesar 0.942, dan F1 score sebesar 0.942. Namun CNN membutuhkan waktu eksekusi paling lama dalam melakukan training dan testing untuk mendapatkan performa terbaiknya yaitu sebesar 458.49 detik.

Rekomendasi untuk penelitian selanjutnya adalah fitur yang digunakan tidak hanya RGB color saja, namun bisa juga menggunakan ekstraksi fitur seperti edge detection, Principal Component Analysis, Fourier descriptor, dan Independent Component Analysis (Kumar and Bhatia, 2014).

## DAFTAR PUSTAKA

- AGARAP, A.F.M., 2018. Deep Learning using Rectified Linear Units (ReLU). *arXiv*, (1), pp.2–8.
- AJAYI, G., 2018. Multi-class Weather Dataset for Image Classification. 1.
- AN, J., CHEN, Y. AND SHIN, H., 2019. Weather Classification using Convolutional Neural Networks. In: *Proceedings - International SoC Design Conference 2018, ISOCC 2018*. Institute of Electrical and Electronics Engineers Inc.pp.245–246.
- ANON 2016. *Automotive Revolution & Perspective Towards 2030*. *Auto Tech Review*, .
- CHOLLET, F. AND O., 2020. *Keras: the Python deep learning API*. [online] Keras: the Python deep learning API. Available at: <<https://keras.io/>> [Accessed 18 Dec. 2020].
- ÇOLAKOĞLU, H.B., 2019. A generalization of the Minkowski distance and a new definition of the ellipse. [online] Available at: <<http://arxiv.org/abs/1903.09657>> [Accessed 29 Dec. 2020].
- DOKMANIC, I., PARHIZKAR, R., RANIERI, J. AND VETTERLI, M., 2015. Euclidean Distance Matrices: Essential Theory, Algorithms and Applications. *IEEE Signal Processing Magazine*, [online] 32(6), pp.12–30. Available at: <<http://arxiv.org/abs/1502.07541>> [Accessed 29 Dec. 2020].
- ELHOSEINY, M., HUANG, S. AND ELGAMMAL, A., 2015. Weather classification with deep convolutional neural networks. *Proceedings - International Conference on Image Processing, ICIP, 2015-Decem(September)*, pp.3349–3353.
- GOLDMAN, R.N., 1991. More matrices and transformations: Shear and pseudo-perspective. In: *Graphics Gems II*. Elsevier Inc.pp.338–341.
- GOOGLE COLAB, 2020. *Welcome to Colaboratory - Colaboratory*. [online] Getting Started - Introduction. Available at: <<https://colab.research.google.com/notebooks/intro.ipynb>> [Accessed 18 Dec. 2020].
- GOUTTE, C. AND GAUSSIER, E., 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *Lecture Notes in Computer Science*, 3408(April), pp.345–359.
- IBRAHIM, M.R., HAWORTH, J. AND CHENG, T., 2019. WeatherNet: Recognising weather and visual conditions from street-level images using deep residual learning. *ISPRS International Journal of Geo-Information*, 8(12).
- JAVIDI, B., 2002. *Image Recognition and Classification*. *Image Recognition and Classification*. CRC Press.
- KANG, L.W., CHOU, K.L. AND FU, R.H., 2019. Deep learning-based weather image recognition. *Proceedings - 2018 International Symposium on Computer, Consumer and Control, IS3C 2018*, pp.384–387.
- KINGMA, D.P. AND BA, J.L., 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp.1–15.
- KUMAR, G. AND BHATIA, P.K., 2014. A detailed review of feature extraction in image processing systems. In: *International Conference on Advanced Computing and Communication Technologies, ACCT*. Institute of Electrical and Electronics Engineers Inc.pp.5–12.
- LECUN, Y., BOTTOU, L., BENGIO, Y. AND HA, P., 1998. LeNet. *Proceedings of the IEEE*, (November), pp.1–46.
- XIA, J., XUAN, D., TAN, L. AND XING, L., 2020. ResNet15: Weather Recognition on Traffic Road with Deep Convolutional Neural Network. *Advances in Meteorology*, 2020.
- ZEILER, M.D. AND FERGUS, R., 2014. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1), pp.818–833.

*Halaman ini sengaja dikosongkan*