

## OPTIMALISASI HYPER PARAMETER CONVOLUTIONAL NEURAL NETWORKS MENGUNAKAN ANT COLONY OPTIMIZATION

Fian Yulio Santoso<sup>\*1</sup>, Eko Sedyono<sup>2</sup>, Hindriyanto Dwi Purnomo<sup>3</sup>

<sup>1,2,3</sup>Universitas Kristen Satya Wacana, Salatiga

Email: <sup>1</sup>fianyuliosantoso@gmail.com, <sup>2</sup>eko@uksw.edu, <sup>3</sup>hindriyanto.purnomo@uksw.edu

<sup>\*</sup>Penulis Korespondensi

(Naskah masuk: 21 Maret 2023, diterima untuk diterbitkan: 25 April 2024)

### Abstrak

Berbagai bidang, termasuk pertanian dan kesehatan, mengalami masalah klasifikasi citra yang dapat diatasi melalui beberapa metode. Salah satu metode tersebut menggabungkan *convolutional neural networks (CNN)* dengan *deep learning*, tetapi *hyperparameter*, seperti fungsi loss, fungsi aktivasi, dan *optimizers*, memengaruhi kinerjanya. *Hyperparameter* ini memerlukan pengoptimalan, dan metode yang ada, seperti algoritma genetika dan pengoptimalan *ant colony*, dapat digunakan untuk tujuan ini. Pengoptimalan *ant colony* terbukti efektif dalam mengoptimalkan *deep learning*, dan penelitian ini berkontribusi pada penyetapan otomatis berbagai *hyperparameter* menggunakan *ant colony* untuk klasifikasi gambar. Pada penelitian ini menggunakan *dataset MNIST* yang bertujuan untuk mengidentifikasi digit pada citra. *Dataset* yang digunakan terbagi menjadi 2, *dataset* pelatihan dan *dataset* validasi. *Dataset* pelatihan terdiri dari 33.600 gambar, dan *dataset* validasi terdiri dari 8.400 gambar. Hasil menunjukkan bahwa optimasi *ant colony* mencapai akurasi 97,46% dengan data validasi dan 99,69% dengan data pelatihan, yang mengungguli algoritma genetika dengan akurasi masing-masing 94,60% dan 97,59% dengan data validasi dan pelatihan. Selain itu, pengoptimalan *ant colony* membutuhkan waktu 27,94 detik untuk dilatih, sedangkan algoritma genetika membutuhkan 22,25 detik.

**Kata kunci:** *hyperparameter auto-tuning, deep neural network, ant colony, algoritma genetika*

### *Convolutional Neural Networks hyperparameter optimization using Ant Colony Optimization*

#### Abstract

Various fields, including agriculture and health, have encountered image classification problems that can be addressed through several methods. One such method combines convolutional neural networks (CNN) with deep learning, but hyperparameters, such as loss functions, activation functions, and optimizers, influence its performance. These hyperparameters require optimization, and existing methods, such as genetic algorithms and ant colony optimization, can be utilized for this purpose. Ant colony optimization has shown to be effective in optimizing deep learning, and this research contributes to automatic tuning of various hyperparameters using ant colonies for image classification. In this study using the MNIST dataset, which aims to identify the digits in the image. The dataset used is divided into 2, training dataset and validation dataset. The training dataset consists of 33,600 images, and the validation dataset consists of 8,400 images. The results indicate that ant colony optimization achieves an accuracy of 97.46% with validation data and 99.69% with training data, which outperforms genetic algorithms with an accuracy of 94.60% and 97.59% with validation and training data, respectively. Additionally, ant colony optimization takes 27.94 seconds to train, while the genetic algorithm requires 22.25 seconds.

**Keywords:** *auto-tuning hyperparameter, deep neural network, ant colony, genetic algorithm*

## 1. PENDAHULUAN

Upaya telah dilakukan untuk mengatasi masalah klasifikasi gambar melalui berbagai pendekatan, seperti teknik pembelajaran mesin termasuk *support vector machines (SVM)* dan *random forests*, dengan tujuan untuk mencapai hasil yang lebih baik

(Murugan, Nair and Kumar, 2019). Banyak studi menggunakan pendekatan *Deep learning*, bersamaan dengan *machine learning*, untuk memecahkan masalah klasifikasi citra karena *Deep learning* telah terbukti sangat efektif dalam tugas ini (Wang, Fan and Wang, 2021). *deep learning* telah terbukti efektif

dalam memecahkan banyak masalah klasifikasi gambar di berbagai domain, termasuk bidang perawatan kesehatan. Misalnya, *deep learning* dapat digunakan untuk mengklasifikasikan COVID-19 berdasarkan gambar *CT scan* paru. Selain itu, *deep learning* dapat membantu dalam mengatasi masalah klasifikasi dalam penginderaan jauh. Di bidang pertanian, ini dapat digunakan untuk mendeteksi penyakit dan hama tanaman.

*deep learning* mampu mengklasifikasikan berbagai jenis masalah karena dapat mempelajari karakteristik penting dari data gambar. Fitur-fitur penting dari suatu gambar dapat diekstraksi menggunakan *convolutional neural network (CNN)* (He et al., 2020). Efektivitas *deep learning* tidak hanya dipengaruhi oleh fitur signifikan yang diperoleh dari citra yang diekstraksi, tetapi banyak faktor yang dapat memengaruhi kinerja *deep learning*. Salah satu faktor tersebut adalah fungsi *loss* yang digunakan. Penelitian Ali Akbari telah menunjukkan bahwa memilih fungsi *loss* yang tepat sangat penting dalam meningkatkan kinerja *deep learning* (Akbari et al., 2021). Selain aspek fungsi *loss*, kuantitas dan keragaman data yang digunakan dalam *deep learning* dapat memengaruhi kinerja *deep learning*. Menurut studi yang dilakukan oleh S. Vani, pemilihan *optimizer* juga dapat memengaruhi kinerja *deep learning* (Vani and Rao, 2019). Selanjutnya, faktor kunci untuk mendapatkan hasil kinerja yang baik adalah para ahli membuat pemodelan atau arsitektur yang sesuai, karena model atau arsitektur yang digunakan dapat memberikan dampak yang signifikan (Cai, Gao and Zhao, 2020). fungsi *loss*, fungsi aktivasi, *optimizer*, dan faktor yang disebutkan sebelumnya adalah *hyperparameter*. Studi ini bertujuan untuk mengonfigurasi *hyperparameter* model *deep learning* secara otomatis, sehingga tidak perlu mencoba semua *hyperparameter* potensial yang tersedia.

## 2. STUDI LITERATUR

Teknik pengoptimalan dapat digunakan untuk menyetel otomatis *hyperparameter* dalam model *deep learning*. Ada berbagai metode yang dapat digunakan dalam teknik optimasi. Salah satunya adalah optimasi koloni semut atau *ant colony*. Pengoptimalan ini terinspirasi dari perilaku semut yang melacak jejak feromon semut lain (Dorigo and Stützle, 2019). Dengan optimasi *ant colony* terbukti dapat bekerja sama dengan *deep learning* dalam menyelesaikan berbagai permasalahan, seperti *Human gait recognition (HGR)* yang bertujuan untuk menjadi alternatif pilihan teknik biometri (Khan et al., 2022). Selain biometri, bidang kesehatan juga menggunakan *deep learning* dan optimasi *ant colony* untuk mengklasifikasikan berbagai penyakit perut (Attique Khan et al., 2021). Contoh lain di bidang pertanian menggunakan pengoptimalan *ant colony* dan *deep learning* untuk membuat sistem rekomendasi prediksi panen (Mythili and Rangaraj,

2021). Semua percobaan yang disebutkan di atas telah membuktikan bahwa menggunakan teknik pengoptimalan *ant colony* dan menggabungkannya dengan *deep learning* terbukti mendapatkan hasil yang luar biasa.

Dalam percobaan oleh Khan Awais dalam membuat alternatif biometri menggunakan *Human gait recognition (HGR)* dengan memanfaatkan optimasi *ant colony* untuk meningkatkan ekstraksi fitur dari video yang dikombinasikan dengan *deep learning* (Khan et al., 2022). Sedangkan pada percobaan yang dilakukan oleh MA Khan dalam mengklasifikasikan penyakit lambung menggunakan *deep learning* dan optimasi *ant colony* untuk *pre-processing* dan *image filtering* (Attique Khan et al., 2021). Kemudian dilakukan penelitian lain oleh Micheal Olaolu Arowolo dalam mendeteksi malaria menggunakan *analysis of variance (ANOVA)* dan *ant colony optimization (ACO)* sebagai seleksi fitur hibrid dalam pemilihan gen yang relevan (Arowolo et al., 2022).

Setelah mengetahui bahwa *ant colony* dapat meningkatkan performa *deep learning* dalam berbagai kasus, perlu diketahui bahwa banyak hal yang mempengaruhi performa *deep learning*. Salah satu contohnya adalah konfigurasi model (Fujiyoshi, Hirakawa and Yamashita, 2019). Hal-hal lain yang dapat memengaruhi performa selain konfigurasi model, seperti pengoptimalan yang digunakan dalam melatih model *deep learning*. Ada berbagai jenis *optimizer* yang tersedia, seperti *Adam*, *RMSprop*, *SGD*, *Adadelta*, *Adagrad*, dan *Adamax*, masing-masing dengan kekuatan dan kelemahannya sendiri. Misalnya, *Stochastic Gradient Descent (SGD)* adalah *optimizer* yang digunakan di banyak bidang, mulai dari pembelajaran mesin dan data besar (Newton, Pasupathy and Yousefian, 2018a) hingga pembelajaran yang mendalam (Zaheer and Shaziya, 2019). *SGD* adalah pilihan yang populer karena struktur iteratifnya yang sederhana, yang cocok untuk memecahkan masalah optimasi (Newton, Pasupathy and Yousefian, 2018b). Sebuah studi oleh Avinash Kumar dalam mendeteksi malaria menemukan bahwa menggunakan pengoptimalan *SGD* menghasilkan akurasi kinerja tinggi sebesar 95%. Penelitian ini juga menunjukkan bahwa pilihan *optimizer* dapat berdampak signifikan pada performa model dan kumpulan data tertentu (Kumar, Sarkar and Pradhan, 2020).

Kumpulan data yang digunakan untuk melatih model dapat berdampak signifikan pada efektivitas *deep learning*, terutama ketika data yang tersedia terbatas. Selain konfigurasi model dan *optimizer*, variasi dan ukuran kumpulan data memiliki dampak pada efektivitas *deep learning*.

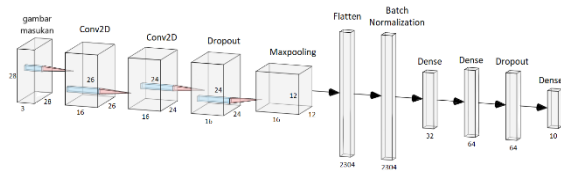
Studi ini menggunakan *dataset* MNIST, yang biasa digunakan untuk tugas pengenalan gambar, untuk mengidentifikasi digit dalam gambar (Colianni, 2017). *Dataset* terdiri dari 42.000 gambar dengan 10 kelas mulai dari 0 hingga 9. *Dataset* dibagi menjadi

dua bagian: *dataset* pelatihan dan *dataset* validasi. *Dataset* pelatihan berisi 33.600 gambar, yang merupakan 80% dari total data, sedangkan *dataset* validasi berisi 8.400 gambar, yang merupakan 20% sisanya dari total data.

Setelah mengetahui berbagai kegunaan optimisasi *ant colony* yang dikombinasikan dengan *deep learning* dan mengetahui berbagai *hyperparameter* yang dapat mempengaruhi kinerja *deep learning*, bab selanjutnya akan memberikan implementasi dan penjelasan penerapan optimisasi *ant colony* untuk *tuning hyperparameter* otomatis.

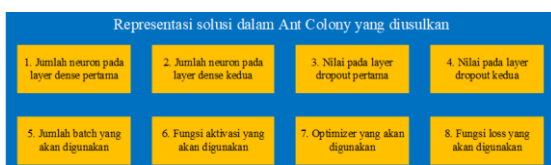
### 3. MODEL YANG DIUSULKAN

Sebelum proses pelatihan, *hyperparameter* disetel secara otomatis. Penelitian ini menggabungkan *Convolutional Neural Networks (CNN)* dengan *ant colony*. Untuk meningkatkan kinerja CNN, lapisan *Dropout* digunakan untuk menghindari *overfitting*, dan *Normalisasi Batch* diadopsi untuk mempercepat proses pelatihan.



Gambar 1. Model Dasar

Arsitektur yang digunakan dalam eksperimen ini bisa dilihat pada Gambar 1. Model diawali dengan pengambilan citra masukan dengan dimensi panjang 28 piksel, lebar 28 piksel, dan kedalaman 3 piksel. Selanjutnya citra diproses melalui dua *convolutional layer*, diikuti oleh *dropout layer* yang nilainya ditentukan oleh *ant colony*, dan kemudian *max pooling layer*. Gambar tersebut kemudian diratakan menjadi bentuk vektor dan melewati lapisan normalisasi *batch*, dua *layer dense*, jumlah neuron ditentukan oleh *ant colony*, dan lapisan *dropout* lainnya yang nilainya ditentukan oleh *ant colony*. Setelah proses ini, model mampu mengklasifikasikan citra ke dalam sepuluh kelas yang tersedia, dengan lapisan padat berisi sepuluh neuron yang terletak di ujung model. Rincian lebih lanjut tentang proses penyetalan *hyperparameter* otomatis menggunakan metode *ant colony* dapat ditemukan di bawah.



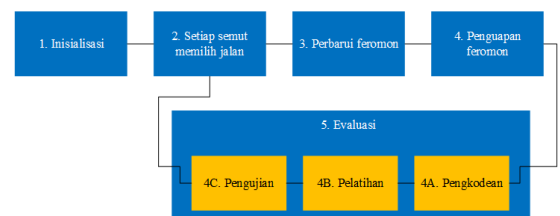
Gambar 2. Pengkodean yang Diusulkan Dengan *Ant Colony*

Penggambaran model pengkodean yang dibuat oleh algoritma *ant colony* untuk mengaktifkan penyetalan *hyperparameter* otomatis ditunjukkan

pada Gambar 2. Model ini mencakup delapan *hyperparameter* yang dapat disetel, dengan informasi lebih lanjut dan kemungkinan pilihan tersedia di Tabel 1.

Tabel 1. Delapan *Hyperparameter* yang Dapat Disetel dan Opsi Tersedia Disetiap *Hyperparameter*

<i>Hyperparameter</i>	Opsi
Jumlah neuron di <i>layer dense</i> pertama	16, 32, 64, 128, 256, 512, 1024, 2048
Jumlah neuron di <i>layer dense</i> kedua	16, 32, 64, 128, 256, 512, 1024, 2048
Nilai pada lapisan <i>dropout</i> pertama	0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8
Nilai pada lapisan <i>dropout</i> kedua	0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8
Jumlah <i>batch</i> yang akan digunakan	16, 32, 64, 128, 256, 512, 1024, 2048
Fungsi aktivasi yang akan digunakan	<i>relu</i> , <i>sigmoid</i> , <i>softplus</i> , <i>softsign</i> , <i>tanh</i> , <i>selu</i> , <i>gelu</i> , <i>linier</i>
<i>Optimizer</i> yang akan digunakan	<i>Adam</i> , <i>RMSprop</i> , <i>SGD</i> , <i>Adadelta</i> , <i>Adagrad</i> , <i>Adamax</i> , <i>Ftrl</i> , <i>Nadam</i>
Fungsi <i>loss</i> yang akan digunakan	<i>Sparse Categorical Crossentropy</i> , <i>Categorical Crossentropy</i> , <i>Binary Crossentropy</i> , <i>Mean Absolute Error</i> , <i>Mean Squared Error</i> , <i>Squared Hinge</i> , <i>CategoricalHinge</i> , <i>CosineSimilarity</i>



Gambar 3. Arsitektur Lengkap *Autotuning Hyperparameter*

Gambar 3 merupakan visualisasi seluruh *ant colony* yang dilakukan pada percobaan ini supaya dapat melakukan *tuning hyperparameter* secara otomatis. Ada lima langkah utama dengan *ant colony*, mulai dari tahap inisialisasi. Pada tahap inisialisasi ditentukan dalam 1 rute yang terdiri dari 8 *hyperparameter*. Ditentukan 20 semut akan melalui rute yang telah dibuat, kemudian nilai evaporasi

ditetapkan sebesar 0.25, kemudian kita mulai dengan membuat sekumpulan solusi yang disebut rute. Semut akan melintasi rute ini. Penelitian ini terdiri dari 8 *hyperparameter* untuk setiap solusi atau rute yang dibuat. Kemudian setiap semut akan melalui rute atau solusi yang telah dibuat. Pada tahap ini, 20 semut akan melewati jalur yang telah dibuat. Pada tahap selanjutnya, setelah semut melewati jalur yang telah dipilih, feromon akan ditambahkan pada jalur yang telah dilalui. Kemudian dilakukan tahap evaporasi feromon untuk laju evaporasi sesuai dengan nilai yang telah ditetapkan pada tahap inisialisasi. Proses pemberian dan penguapan feromon sangat penting karena akan digunakan untuk membuat rute yang harus diikuti semut berikutnya, kemudian dilakukan tahap evaluasi. Tahap evaluasi terdiri dari 3 tahapan utama yaitu pengkodean, pelatihan, dan pengujian. Pada tahap pengkodean, solusi yang dibuat di *ant colony* diterjemahkan ke dalam *hyperparameter* yang telah dijelaskan sebelumnya. Setelah berhasil menerjemahkan, dilakukan pelatihan dengan *hyperparameter* yang dibuat dengan *ant colony*, dan dilakukan pengujian. Hasil yang diperoleh pada tahap sebelumnya akan disimpan, dan hasil terbaik yang diperoleh akan digunakan sebagai solusi.

#### 4. HASIL EKSPERIMEN

Tabel 2 menampilkan lingkungan yang digunakan dalam melakukan penelitian ini.

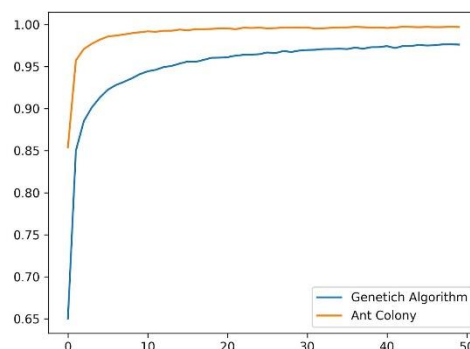
Tabel 1. Lingkungan yang Digunakan

Hardware/Software	Spesifikasi/Versi
Prosesor	AMD EPYC 7742
vCPU	16 Core
vRAM	128 GB
GPU	1xNVIDIA A100-SXM
Versi Python	3.8.10
Versi Tensorflow	2.8.0

Gambaran komprehensif tentang lingkungan penelitian dapat ditemukan pada Tabel 2. Replikasi penelitian ini dengan lingkungan yang berbeda dapat menghasilkan hasil yang berbeda, menggarisbawahi pentingnya informasi ini.

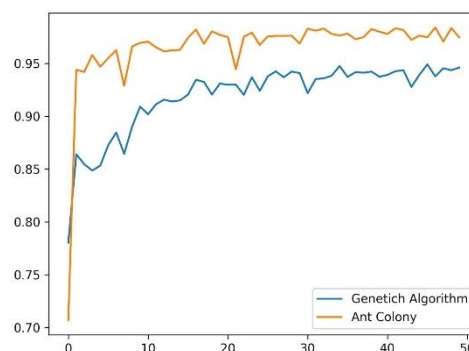
Studi ini membandingkan *ant colony* dengan algoritma genetika untuk melakukan penyesuaian *hyperparameter* secara otomatis. Model dasar kedua metode, kumpulan data, dan opsi *hyperparameter*nya sama untuk perbandingan yang adil. Untuk pengaturan algoritma genetika, ada 10 solusi dalam 1 populasi. Untuk metode pemilihan induk terbaik dibuat solusi baru dengan menggunakan metode *Roulette Wheel Selection (RWS)*. Untuk melakukan *crossover* menggunakan metode titik tunggal.

Sedangkan untuk melakukan mutasi menggunakan metode *swap*.



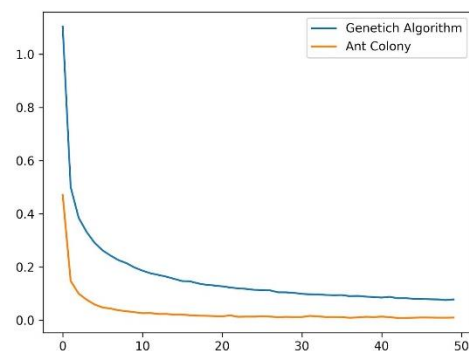
Gambar 4. Perbandingan Akurasi Dengan Data Pelatihan

Gambar 4 mengilustrasikan perbandingan akurasi yang diperoleh dengan memanfaatkan *dataset* pelatihan pada model dengan *hyperparameter* optimal. Teknik *ant colony* mencapai tingkat akurasi yang luar biasa sebesar 99,69%, sedangkan algoritma genetika hanya mencapai akurasi sebesar 97,59%.



Gambar 5. Perbandingan akurasi dengan data validasi

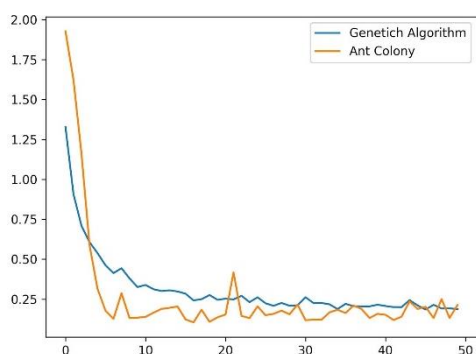
Gambar 5 menyajikan perbandingan hasil akurasi yang diperoleh dari *dataset* validasi pada model yang memiliki *hyperparameter* paling optimal. Algoritma *ant colony* mencapai skor akurasi 97,46%, sedangkan algoritma genetika memperoleh skor akurasi lebih rendah yaitu 94,60%.



Gambar 6. Perbandingan Loss Dengan Data Pelatihan

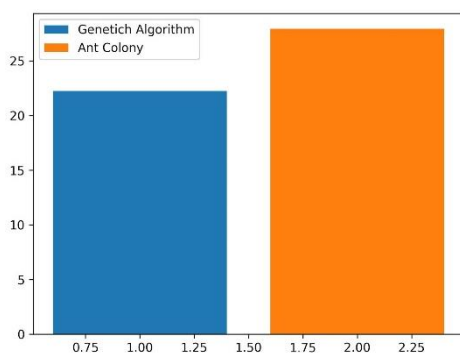
Gambar 6 menampilkan perbandingan *loss* dengan *dataset* pelatihan antara model dengan

*hyperparameter* terbaik. Pendekatan *ant colony* menghasilkan nilai *loss* sebesar 0,0094, sedangkan metode algoritma genetika menghasilkan nilai *loss* sebesar 0,0777.



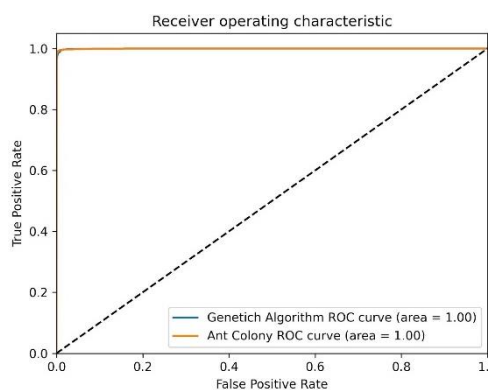
Gambar 7. Perbandingan Loss Dengan Data Validasi

Gambar 7 menampilkan perbandingan nilai *loss* yang dicapai model dengan *hyperparameter* optimal menggunakan *dataset* validasi. Algoritma *ant colony* mendapatkan nilai *loss* sebesar 0.2130, sedangkan algoritma genetika mendapatkan nilai *loss* sebesar 0.1874.



Gambar 8. Perbandingan Waktu Pelatihan

Perbandingan waktu pelatihan yang dibutuhkan dalam detik untuk model dengan *hyperparameter* terbaik diilustrasikan pada Gambar 8. Metode *ant colony* membutuhkan waktu 27,94 detik, sedangkan algoritma genetika membutuhkan waktu 22,25 detik.



Gambar 9. Perbandingan ROC

Nilai *ROC* terbaik pada model dengan *hyperparameter* terbaik diperoleh *ant colony* seperti pada Gambar 9, melebihi algoritma genetika.

## 5. KESIMPULAN

*Ant colony* digunakan untuk melakukan pengaturan *hyperparameter* secara otomatis untuk klasifikasi citra pada penelitian ini. Dibandingkan dengan algoritma genetika, *ant colony* memberikan hasil terbaik dalam hal akurasi, namun untuk kecepatan waktu pelatihan sedikit lebih lama dibandingkan dengan algoritma genetika.

## 6. UCAPAN TERIMA KASIH

Pendanaan penelitian ini diberikan oleh Universitas Kristen Satya Wacana melalui Grand No 190/Pen./Rek./6/V/2022.

## DAFTAR PUSTAKA

- AKBARI, A., AWAIS, M., BASHAR, M. AND KITTLER, J., 2021. How Does Loss Function Affect Generalization Performance of Deep Learning? Application to Human Age Estimation. In: International Conference on Machine Learning. [online] International Conference on Machine Learning. PMLR. pp.141–151. Available at: <<https://proceedings.mlr.press/v139/akbari21a.html>> [Accessed 14 March 2022].
- AROWOLO, M.O., AWOTUNDE, J.B., AYEGBA, P. AND SULYMAN, S.O.H., 2022. Relevant gene selection using ANOVA-ant colony optimisation approach for malaria vector data classification. International Journal of Modelling, Identification and Control, [online] 41(1/2), p.12. <https://doi.org/10.1504/ijmic.2022.127093>.
- ATTIQUE KHAN, M., MAJID, A., HUSSAIN, N., ALHAISONI, M., ZHANG, Y.-D., KADRY, S. AND NAM, Y., 2021. Multiclass stomach diseases classification using deep learning features optimization. Computers, materials & continua, [online] 67(3), pp.3381–3399. <https://doi.org/10.32604/cmc.2021.014983>.
- CAI, L., GAO, J. AND ZHAO, D., 2020. A review of the application of deep learning in medical image classification and segmentation. Annals of translational medicine, [online] 8(11), p.713. <https://doi.org/10.21037/atm.2020.02.44>.
- COLIANNI, S., 2017. MNIST as .jpg. Available at: <<https://www.kaggle.com/scolianni/mnistasjpg>> [Accessed 28 October 2022].
- DORIGO, M. AND STÜTZLE, T., 2019. Ant colony optimization: Overview and recent advances. In: Handbook of Metaheuristics,

- International series in operations research & management science. [online] Cham: Springer International Publishing. pp.311–351. [https://doi.org/10.1007/978-3-319-91086-4\\_10](https://doi.org/10.1007/978-3-319-91086-4_10).
- FUJIYOSHI, H., HIRAKAWA, T. AND YAMASHITA, T., 2019. Deep learning-based image recognition for autonomous driving. *IATSS research*, [online] 43(4), pp.244–252. <https://doi.org/10.1016/j.iatssr.2019.11.008>.
- HE, T., LIU, Y., YU, Y., ZHAO, Q. AND HU, Z., 2020. Application of deep convolutional neural network on feature extraction and detection of wood defects. *Measurement*, [online] 152(107357), p.107357. <https://doi.org/10.1016/j.measurement.2019.107357>.
- KHAN, A., ATTIQUE KHAN, M., YOUNUS JAVED, M., ALHAISONI, M., TARIQ, U., KADRY, S., CHOI, J.-I. AND NAM, Y., 2022. Human gait recognition using deep learning and improved ant colony optimization. *Computers, materials & continua*, [online] 70(2), pp.2113–2130. <https://doi.org/10.32604/cmc.2022.018270>.
- KUMAR, A., SARKAR, S. AND PRADHAN, C., 2020. Malaria disease detection using CNN technique with SGD, RMSprop and ADAM optimizers. In: *Studies in Big Data*. [online] Cham: Springer International Publishing. pp.211–230. [https://doi.org/10.1007/978-3-030-33966-1\\_11](https://doi.org/10.1007/978-3-030-33966-1_11).
- MURUGAN, A., NAIR, S.A.H. AND KUMAR, K.P.S., 2019. Detection of Skin Cancer Using SVM, Random Forest and kNN Classifiers. *Journal of medical systems*, [online] 43(8), p.269. <https://doi.org/10.1007/s10916-019-1400-8>.
- MYTHILI, K. AND RANGARAJ, R., 2021. Crop Recommendation for Better Crop Yield for Precision Agriculture Using Ant Colony Optimization with Deep Learning Method. *Annals of the Romanian Society for Cell Biology*, [online] pp.4783–4794. Available at: <<https://www.annalsofrscb.ro/index.php/journal/article/view/3024>> [Accessed 15 February 2023].
- NEWTON, D., PASUPATHY, R. AND YOUSEFIAN, F., 2018a. Recent trends in stochastic gradient descent for machine learning and big data. In: *2018 Winter Simulation Conference (WSC)*. [online] 2018 Winter Simulation Conference (WSC). IEEE. <https://doi.org/10.1109/wsc.2018.8632351>.
- NEWTON, D., PASUPATHY, R. AND YOUSEFIAN, F., 2018b. Recent trends in stochastic gradient descent for machine learning and big data. In: *2018 Winter Simulation Conference (WSC)*. [online] 2018 Winter Simulation Conference (WSC). IEEE. <https://doi.org/10.1109/wsc.2018.8632351>.
- VANI, S. AND RAO, T.V.M., 2019. An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. [online] 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE. <https://doi.org/10.1109/icoei.2019.8862686>.
- WANG, P., FAN, E.N. AND WANG, P., 2021. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern recognition letters*, [online] 141, pp.61–67. <https://doi.org/10.1016/j.patrec.2020.07.042>.
- ZAHEER, R. AND SHAZIYA, H., 2019. A study of the optimization algorithms in deep learning. In: *2019 Third International Conference on Inventive Systems and Control (ICISC)*. [online] 2019 Third International Conference on Inventive Systems and Control (ICISC). IEEE. <https://doi.org/10.1109/icisc44355.2019.9036442>.