

**KLASIFIKASI LARVA NYAMUK AEDES MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORKS DAN *EXTREME
LEARNING MACHINE***

SKRIPSI

IVAN JOSHUA

191401084



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2023

KLASIFIKASI LARVA NYAMUK AEDES MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORKS DAN *EXTREME*
LEARNING MACHINE

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer

IVAN JOSHUA

191401084



PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

MEDAN

2023

PERSETUJUAN

Judul : KLASIFIKASI LARVA NYAMUK AEDES
MENGUNAKAN CONVOLUTIONAL
NEURAL NETWORKS DAN EXTREME
LEARNING MACHINE

Kategori : SKRIPSI

Nama : IVAN JOSHUA

Nomor Induk Mahasiswa : 191401084

Program Studi : SARJANA (S1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI UNIVERSITAS SUMATERA
UTARA

Telah diuji dan dinyatakan lulus di Medan, 6 Desember 2023

Dosen Pembimbing II

Dosen Pembimbing I

Fuzy Yustika Manik S.Kom., M.Kom

NIP. 198710152019032010

Pauzi Ibrahim Nainggolan, S.Komp., M.Sc.

NIP. 198809142020011001

Diketahui/Disetujui Oleh
Ketua Program Studi S-1 Ilmu Komputer

Dr. Amalia S.T., M.T.

NIP. 197812212014042001

PERNYATAAN**KLASIFIKASI LARVA NYAMUK AEDES MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORKS DAN EXTREME
LEARNING MACHINE****SKRIPSI**

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

Medan,



Ivan Joshua

191401084

UCAPAN TERIMA KASIH

Puji syukur kehadiran Tuhan Yang Maha Esa atas anugerah, berkat dan karunia yang telah dilimpahkan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini yang merupakan salah satu persyaratan untuk memperoleh gelar Sarjana Komputer dari program studi S-1 Ilmu Komputer Universitas Sumatera Utara.

Pada kesempatan ini penulis ingin memberikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah mendukung dan membantu pengerjaan skripsi ini. Penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang senantiasa menyertai dan memberikan berkat Kesehatan kepada penulis baik dalam pembuatan skripsi ini.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara
3. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dr. Amalia S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Fauzan Nurahmadi S.Kom., M.Cs selaku Dosen Pembimbing Akademik penulis di Program Studi S-1 Ilmu Komputer.
6. Bapak Pauzi Ibrahim Nainggolan, S.Komp., M.Sc. selaku Dosen Pembimbing I yang telah membimbing penulis dalam penyusunan skripsi ini.
7. Ibu Fuzy Yustika Manik S.Kom., M.Kom selaku Dosen Pembimbing II yang telah membimbing dan memberikan saran dan masukan kepada penulis dalam penyusunan skripsi ini.
8. Bapak/Ibu staff pengajar serta seluruh pegawai Program Studi S-1 Ilmu Komputer Fasilkom-TI USU.
9. Orang tua dan keluarga yang selalu mendoakan dan memberikan dukungan baik dari segi moral maupun material.
10. Sahabat-sahabat saya Celine D, Christine A, Filbert, Harris K, Jasmine P, Nicholas, Nicholas W, Niko D, Prisko B, Ricky A, Sinu, Valencia X, Vinny A yang telah menemani penulis dalam proses penulisan skripsi ini.

11. Serta seluruh pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat disebutkan secara rinci.

Demikianlah skripsi ini dibuat, semoga bermanfaat bagi pembaca pada umumnya dan kepada penulis khususnya. Penulis menyadari bahwa masih banyak terdapat kekurangan yang ada di dalam skripsi ini. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak yang membaca skripsi ini dan untuk itu penulis mengucapkan terima kasih.

Medan,



Penulis

ABSTRAK

Penyebaran penyakit yang ditularkan oleh nyamuk *Aedes* seperti demam berdarah dengue (DBD), Chikungunya (CHIKV), dan zika merupakan masalah kesehatan yang serius di banyak negara. Nyamuk *Aedes* dapat ditemukan di daerah tropis dan subtropis. Salah satu metode untuk mengontrol penyakit-penyakit tersebut adalah dengan mengendalikan populasi nyamuk *Aedes*, dan salah satu cara untuk mengendalikan populasi nyamuk *Aedes* adalah dengan membasmi larva nyamuk *Aedes*. Untuk mengidentifikasi jenis larva nyamuk diperlukan waktu dan tenaga yang banyak serta keahlian khusus dalam pengenalan morfologi dan anatomi larva nyamuk agar dapat mengklasifikasikan jenis larva nyamuk. Oleh sebab itu, dibutuhkanlah sistem yang dapat mengklasifikasikan larva nyamuk *Aedes* dan *non-Aedes*. Pada penelitian ini, data yang digunakan berasal dari Balai Teknik Kesehatan Lingkungan Dan Pengendalian Penyakit (BTKLPP) Kelas I Medan, klasifikasi dilakukan dengan menggunakan gambar larva yang diambil menggunakan mikroskop digital. Metode yang digunakan untuk mengklasifikasikan gambar larva nyamuk adalah *Convolutional Neural Networks* (CNN) yang berfungsi untuk mengekstraksi fitur dari gambar larva dan *Extreme Learning Machine* (ELM) untuk mengklasifikasikan gambar larva. Hasil yang didapat dari evaluasi model CNN-ELM adalah akurasi 98%, *f1-score Aedes* 99%, dan *f1-score non-Aedes* 96%.

Kata Kunci : *Aedes*, *Convolutional Neural Networks*, *Extreme Learning Machine*, klasifikasi, *Machine Learning*

*CLASSIFICATION OF AEDES MOSQUITO LARVA USING
CONVOLUTIONAL NEURAL NETWORKS AND
EXTREME LEARNING MACHINE*

ABSTRACT

The spread of diseases transmitted by Aedes mosquitoes, such as Dengue fever, Chikungunya (CHIKV), and zika, is a serious health problem in many countries. Aedes mosquitoes can be found in tropical and subtropical areas. One method to control the diseases is by controlling the population of Aedes mosquitoes, and one way to control the population of Aedes mosquito is by exterminating Aedes mosquitoes larvae. However, to identify mosquito larvae requires a significant amount of effort, time and specialized knowledge about the morphology and anatomy of mosquitoes larvae in order to classify the types of mosquito larvae. Therefore, a system is needed to classify Aedes and non-Aedes mosquito larvae. In this study, the data used in this study is provided by Center for Environmental Health Engineering and Disease Control (BTKLPP) Class 1 Medan, classification was carried out using larvae images taken using a digital microscope. The methods used to classify mosquito larvae images are Convolutional Neural Networks (CNN) to extract feature from the larvae image and Extreme Learning Machine(ELM) to classify the larvae image. The result obtained from evaluating the CNN-ELM model is an accuracy of 98%, f1-score of 99% for Aedes larvae and and f1-score of 96% for non-Aedes larvae.

Keywords: *Aedes, Convolutional Neural Networks, Extreme Learning Machine, Classification, Machine Learning*

DAFTAR ISI

PERSETUJUAN.....	ii
PERNYATAAN.....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. <i>Machine Learning</i>	6
2.1.1. <i>Neural Network</i>	6
2.1.2. <i>Convolutional Neural Networks (CNN)</i>	6
2.1.3. <i>Extreme Learning Machine (ELM)</i>	9
2.2. Nyamuk <i>Aedes</i>	10
2.3. Klasifikasi Gambar.....	11
2.4. Optuna.....	11
2.4.1. <i>Hyperparameter</i>	12
2.5. <i>Confusion Matrix</i>	12
2.6. <i>Activation Functions</i>	14
2.6.1. <i>Sigmoid Function</i>	15
2.6.2. <i>Hyperbolic Tangent (tanh)</i>	15
2.6.3. <i>Rectified Linear Unit (ReLU)</i>	16
2.7. <i>Normalization</i>	16

2.8.	<i>Luminosity</i>	16
BAB III ANALISIS DAN PERANCANGAN SISTEM		17
3.1.	Analisis Sistem	17
3.1.1.	<i>Analisis masalah</i>	17
3.1.2.	<i>Analisis kebutuhan</i>	18
3.2.	Penjelasan Penelitian	19
3.2.1.	<i>Sumber Dataset</i>	20
3.2.2.	<i>Pra-proses gambar</i>	20
3.2.3.	<i>Ekstraksi Fitur dengan CNN</i>	21
3.2.4.	<i>Klasifikasi dengan ELM</i>	21
3.2.5.	<i>Evaluasi model</i>	21
BAB IV IMPLEMENTASI DAN PENGUJIAN		22
4.1.	Perangkat yang Digunakan	22
4.1.1.	<i>Perangkat Keras</i>	22
4.1.2.	<i>Perangkat Lunak</i>	22
4.1.3.	<i>Library</i>	22
4.2.	Penerapan Tahap Pengumpulan Dataset	23
4.3.	Penerapan Tahap Pra-proses Dataset	23
4.4.	Penerapan Tahap Ekstraksi Fitur dengan CNN	28
4.5.	Penerapan Tahap Klasifikasi dengan ELM	31
4.6.	Penerapan Tahap Evaluasi model	35
BAB V KESIMPULAN DAN SARAN		37
5.1.	Kesimpulan	37
5.2.	Saran	37
DAFTAR PUSTAKA		38

DAFTAR GAMBAR

Gambar 2.1 Struktur CNN.....	7
Gambar 2.2 Convolution layer	8
Gambar 2.3 Max Pooling layer.....	9
Gambar 2.4 Average Pooling layer	9
Gambar 2.5 Struktur ELM.....	10
Gambar 2.6 Morfologi Larva Aedes.....	11
Gambar 2.7 Confusion Matrix.....	13
Gambar 3.1 Diagram Ishikawa (Fishbone Diagram).....	17
Gambar 3.2 Alur Penelitian	20
Gambar 4.1 Contoh Gambar Larva, Aedes (atas), non-Aedes (bawah)	23
Gambar 4.2 Gambar Kode tahap Pra-proses	24
Gambar 4.3 Gambar Larva Aedes yang di Pra-proses	24
Gambar 4.4 Gambar Larva Aedes yang di Pra-proses	25
Gambar 4.5 Gambar Larva Aedes yang di Pra-proses	25
Gambar 4.6 Gambar Larva Aedes yang di Pra-proses	25
Gambar 4.7 Gambar Larva Aedes yang di Pra-proses	26
Gambar 4.8 Gambar Larva Non-Aedes yang di Pra-proses	26
Gambar 4.9 Gambar Larva Non-Aedes yang di Pra-proses	26
Gambar 4.10 Gambar Larva Non-Aedes yang di Pra-proses	27
Gambar 4.11 Gambar Larva Non-Aedes yang di Pra-proses	27
Gambar 4.12 Gambar Larva Non-Aedes yang di Pra-proses	27
Gambar 4.13 Gambar Kode Model CNN.....	29
Gambar 4.14 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar Aedes (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)	29
Gambar 4.15 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar Non- Aedes (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut).....	30
Gambar 4.16 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar Aedes (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)	30
Gambar 4.17 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar Non- Aedes (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut).....	31
Gambar 4.18 Gambar Kode Optuna	34
Gambar 4.19 Gambar Kode Model ELM.....	34
Gambar 4.20 Gambar Kode Evaluasi Model.....	36

DAFTAR TABEL

Tabel 4.1 Hyperparameter diperoleh Optuna dengan CNN belum dilatih.. 32

Tabel 4.2 Hyperparameter diperoleh Optuna dengan CNN sudah dilatih... 32

BAB I

PENDAHULUAN

1.1. Latar Belakang

Nyamuk *Aedes* dapat menyebarkan beberapa penyakit seperti demam berdarah *dengue* (DBD), *Chikungunya* (CHIKV) dan *zika*, nyamuk dengan spesies *Aedes Aegypti*. Spesies *Aedes Aegypti* dianggap sebagai penyebab utama transmisi penyakit ini, nyamuk *Aedes* biasa ditemukan di daerah tropis namun dengan terjadinya perubahan iklim global, nyamuk *Aedes* juga dapat ditemukan di beberapa daerah non-tropis (Sanchez-Ortiz, A et al., 2019). Virus *zika* dan demam berdarah *dengue* (DBD) mampu mengubah aroma tubuh manusia dan tikus yang terinfeksi dengan tujuan untuk menarik perhatian nyamuk-nyamuk yang akan menghisap darah yang terinfeksi dan membawa virus ke korban berikutnya (Zhang, H et al., 2020).

Menurut Direktorat Pencegahan dan Pengendalian Penyakit Menular (P2PM) dari Januari 2022 sampai minggu ke 36, jumlah kumulatif kasus konfirmasi DBD yang dilaporkan terdapat sebanyak 87.501 kasus dan 816 kematian (Kementrian Kesehatan Republik Indonesia., 2022). Studi tentang prevalensi *dengue* menyatakan bahwa 3,9 miliar orang berisiko terinfeksi dengan virus *dengue*. Meskipun risiko infeksi ada di 129 negara, 70% risiko terjadi di Asia. Jumlah permasalahan DBD yang dilaporkan ke *World Health Organization* bertambah lebih dari 8 kali lipat dalam 2 dekade terakhir, dari 505.430 permasalahan pada tahun 2000, menjadi lebih dari 2,4 juta pada tahun 2010, serta 5,2 juta pada tahun 2019. Kematian yang dilaporkan antara tahun 2000 serta 2015 bertambah dari 960 menjadi 4032 (*World Health Organization.*, 2020). Oleh karena itu diperlukan cara untuk mengontrol populasi nyamuk *Aedes* sehingga penyebaran penyakit yang disebabkan oleh nyamuk *Aedes* dapat dikurangi.

Dengan perkembangan teknologi seperti *deep learning*, dapat digunakan untuk melawan penyakit yang ditularkan oleh nyamuk *Aedes* dengan mencegah penyebaran nyamuk *Aedes* melalui pembasmian nyamuk *Aedes* saat masih di tahap larva, terdapat metode klasifikasi gambar dengan *deep learning* yang dapat membedakan larva nyamuk *Aedes* dan *Non-Aedes* (Azman, M. I. A. B. Z., & Sarlan,

A. B., 2020). Dalam sistem ini, ciri-ciri fisik larva *Aedes* diambil dengan menggunakan *Convolutional Neural Networks* (CNN), CNN merupakan salah satu metode *deep learning* dan CNN menunjukkan keunggulan dalam mengambil ciri-ciri gambar dibandingkan dengan metode lainnya (Wang, L. et al., 2018), untuk melatih CNN untuk mengklasifikasikan gambar digunakanlah algoritma *back-propagation*, oleh karena itu diperlukanlah data dalam jumlah yang banyak, *graphics card* yang memadai dan waktu yang diperlukan mungkin memerlukan beberapa hari atau minggu (Khellal, A., Ma, H., & Fei, Q., 2018).

Oleh karena itu untuk mengurangi waktu latihan yang lama digunakanlah *Extreme learning machine* (ELM) yang memiliki kelebihan yaitu menjalankan latihan dengan waktu yang cukup cepat dengan memastikan akurasi pembelajaran (Wang, L. et al., 2018). ELM menggunakan metode pembelajaran satu arah yang terdiri atas sebuah layer *input*, sebuah *hidden layer*, dan sebuah layer *output*. Arititektur dari *extreme learning machine* mudah dimengerti dan tidak memerlukan pengaturan parameter secara berulang yang membuat proses pelatihan yang lebih cepat dan mendapatkan performa yang memuaskan dalam klasifikasi (Nahiduzzaman, M. et al., 2021).

Dalam penelitian yang dilakukan oleh Khellal, A., Ma, H., & Fei, Q. (2018), dilakukanlah perbandingan antara sistem yang menggunakan algoritma *Extreme Learning Machine-Convolutional Neural Networks* (ELM-CNN) dengan *Back Propagation-Convolutional Neural Networks* (BP-CNN) yang telah dibuat dengan dua aspek, yaitu kuantitatif dan kualitatif. Dari perbandingan tersebut disimpulkan bahwa ELM-CNN memiliki beberapa keunggulan seperti, lima kali lebih cepat, tidak memerlukan pengaturan parameter secara berulang, performa generalisasi yang kompetitif, algoritma pembelajaran bertahap tanpa pengawasan. Penelitian yang dilakukan oleh De Silva, W. D. M., & Jayalal, S. (2020) menggunakan CNN untuk mengidentifikasi larva *Aedes* dengan menggunakan kamera *smartphone* yang memiliki kamera 12 *mega-pixels*, dan dapat di *zoom* delapan kali secara digital, dan terpasang lensa pembesar dengan perbesaran 60 kali, akurasi yang didapat dengan menggunakan lensa pembesar adalah 97% dan akurasi yang didapat tanpa menggunakan lensa pembesar adalah 77,13%.

Dalam penelitian Kannoja, S. P., & Jaiswal, G. (2018) dilakukan perbandingan akurasi dalam klasifikasi gambar antar model CNN, ELM, gabungan *Convolutional neural networks-Extreme learning machine* (CNN-ELM), dan model kelompok CNN-ELM dengan menggunakan data dari database *Modified National Institute of Standards and Technology database* (MNIST). Hasil akurasi yang didapat oleh model CNN adalah 99,20%, model ELM mendapatkan 97.54%, CNN-ELM mendapatkan 99,24%, dan model kelompok CNN-ELM mendapatkan 99,33%. Penelitian yang dilakukan oleh Sanchez-Ortiz, A. (2017) menggunakan CNN untuk mengidentifikasi larva *Aedes*, hasil yang didapat adalah persentase rata-rata dalam mengidentifikasi larva nyamuk *Aedes* adalah 96,88%, sedangkan persentase rata-rata saat mengidentifikasi larva nyamuk *non-Aedes* adalah 64,95%, hal ini disebabkan oleh sedikitnya data yang digunakan untuk melatih model CNN.

1.2. Rumusan Masalah

Pada penelitian terdahulu, CNN digunakan untuk mengklasifikasikan larva nyamuk *Aedes*, hasil yang didapat saat mengidentifikasi larva nyamuk *Aedes* adalah 96,88% dan saat mengidentifikasi larva nyamuk *non-Aedes* adalah 64,95%. Oleh karena itu penulis mencoba untuk membuat sistem untuk mengklasifikasikan larva nyamuk *Aedes* dengan model CNN-ELM yang diharapkan dapat meningkatkan hasil akurasi klasifikasi larva nyamuk *Aedes* dan *non-Aedes*.

1.3. Batasan Masalah

Dalam penelitian ini terdapat beberapa batasan, yaitu:

1. Hanya dapat membedakan *Aedes* dan *non-Aedes*.
2. Gambar yang digunakan untuk klasifikasi harus beresolusi minimal 640x480 piksel.
3. Sistem dibangun menggunakan bahasa pemrograman Python.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membuat sistem klasifikasi larva nyamuk *Aedes* menggunakan pendekatan *convolutional neural networks* dan *extreme learning machine*.

1.5. Manfaat Penelitian

Penelitian ini dapat memberikan kontribusi penting dalam mengidentifikasi larva nyamuk *Aedes*, sehingga proses pengelompokkan larva dapat dilakukan oleh peneliti secara lebih mudah.

1.6. Metodologi Penelitian

Metodologi penelitian yang digunakan adalah sebagai berikut:

1. Studi Pustaka

Pada tahap ini penelitian dimulai dengan mencari data-data dari berbagai sumber terpercaya dan melakukan peninjauan pustaka melalui buku-buku, jurnal, e-book, artikel ilmiah, makalah ataupun situs internet tentang ciri-ciri larva nyamuk *Aedes*, algoritma Extreme Learning Machine, dan algoritma Convolutional Neural Networks.

2. Analisis dan Perancangan sistem

Berdasarkan ruang lingkup penelitian, penulis melakukan analisa terhadap apa saja yang akan dibutuhkan dalam penelitian untuk segera dirancang dalam sebuah diagram alir (*flowchart*).

3. Implementasi

Pada tahap ini, membuat sebuah sistem dengan menggunakan bahasa pemrograman python sesuai dengan diagram alir yang telah dirancang.

4. Pengujian

Pada tahap ini, sistem yang telah dirancang akan diuji coba untuk mengetahui apakah sistem berjalan sesuai tujuan.

5. Dokumentasi

Pada tahap ini dilakukanlah dokumentasi mulai dari tahap analisa sistem sampai kepada pengujian sistem dalam format penulisan berbentuk skripsi.

1.7. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi dalam beberapa bab, yaitu:

BAB I PENDAHULUAN

Bab pendahuluan membahas mengenai latar belakang penelitian, rumusan masalah, batasan masalah penelitian, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan skripsi.

BAB II LANDASAN TEORI

Bab landasan teori membahas tentang penjelasan mengenai nyamuk *Aedes*, klasifikasi gambar, *Convolutional Neural Networks (CNN)*, *Extreme Learning Machine (ELM)*, dan penelitian-penelitian yang relevan.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab analisis dan perancangan sistem membahas tentang analisis dari masalah-masalah penelitian dan solusi yang diberikan untuk permasalahan tersebut dengan sistem perancangan sistem yang akan dibangun.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab implementasi dan pengujian siste membahas tentang implementasi dari sistem yang telah dirancang dan hasil dari pengujian sistem yang berfungsi untuk memeriksa kesesuaian dan keberhasilan sistem.

BAB V KESIMPULAN DAN SARAN

Bab kesimpulan dan saran berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran dari hasil pengujian sehingga dapat diterapkan menjadi masukan dalam penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1. Machine Learning

Machine learning adalah salah satu cabang dari kecerdasan buatan yang bertujuan untuk mengembangkan algoritma dan teknik-teknik untuk membuat sistem komputer mampu belajar secara mandiri dari data yang ada dan melakukan prediksi atau tindakan berdasarkan hasil pembelajaran tersebut (Angra, S., & Ahuja, S. 2017). *Machine learning* terdiri dari dua jenis pendekatan yaitu *supervised learning* dan *unsupervised learning*. Pada *supervised learning*, algoritma belajar dari data yang telah diberi label atau anotasi sebelumnya, sementara pada *unsupervised learning*, algoritma belajar dari data yang tidak diberi label dan mencoba mengelompokkan data tersebut berdasarkan pola-pola yang terdapat di dalamnya. *Machine learning* telah diterapkan pada berbagai bidang seperti pengolahan citra, pengenalan suara, pengenalan objek, pengenalan pola, *deep learning*, dan lain-lain. (Ebert, C., & Louridas, P., 2016).

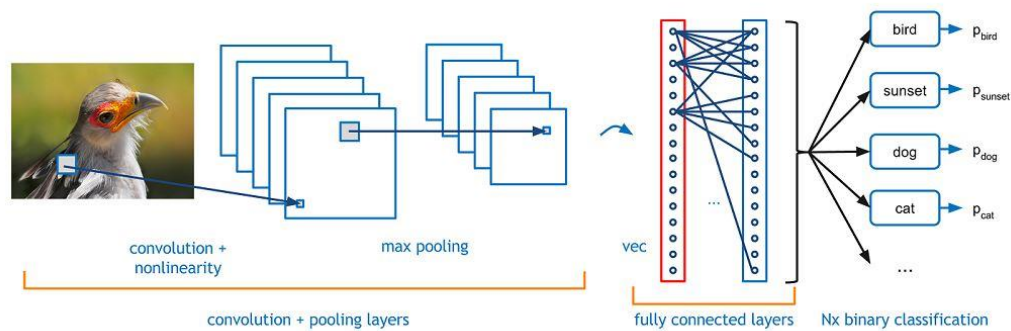
2.1.1. Neural Network

Neural network adalah sebuah metode dalam ilmu komputer dan matematika yang terinspirasi oleh cara kerja otak manusia. *Neural network* memiliki kemampuan untuk belajar dari data dan mengenali pola atau hubungan yang kompleks dalam data tersebut oleh karena itu *neural network* mulai digunakan untuk menyelesaikan masalah seperti pengenalan objek, pengenalan ucapan, prediksi pasar saham dan lain-lain. (Bhutani, K. R., & Farsaie, A. 1991).

2.1.2. Convolutional Neural Networks (CNN)

CNN merupakan bagian dari neural networks, CNN mempunyai tiga komponen, yaitu *convolution layer* (CONV), *pooling layer* (POOL), and *fully connected layers* (FC). CNN dilatih dengan menggunakan metode *back-propagation* yang membutuhkan dataset yang besar, dan GPU yang memadai. CNN menjadi solusi untuk klasifikasi, lokalisasi, dan deteksi, oleh karena itu CNN banyak digunakan di sistem-sistem pengenalan. (Khellal, A., Ma, H., & Fei, Q., 2018). CNN dapat melakukan ekstraksi fitur dan mengenali fitur tersebut dalam waktu yang sama, CNN dapat mengolah data yang berbentuk array multi-dimensi. CNN mengambil

representasi abstrak dari data seperti gambar, suara, teks, video, dan lain-lain, CNN juga memiliki kekurangan, yaitu membutuhkan biaya komputasi yang tinggi (Yoo, Y. & Oh, S.-Y. 2016). Gambar 2.1 merupakan struktur CNN.



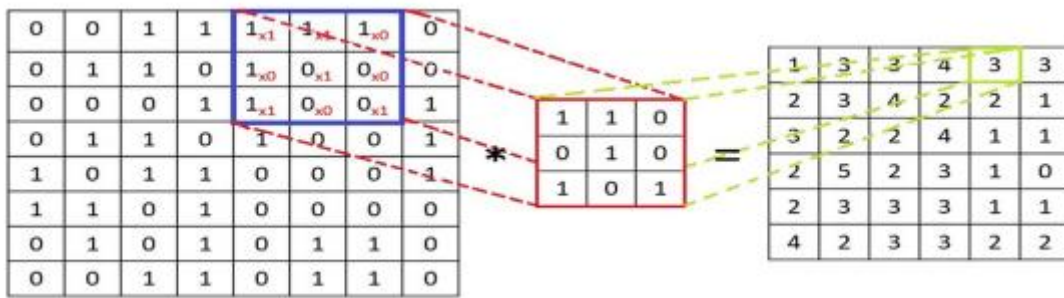
Gambar 2.1 Struktur CNN

2.1.2.1. Convolution Layer

Convolution Layer adalah filter yang diterapkan pada tiap wilayah gambar input secara berkali-kali agar dapat menciptakan *feature map* yang berisi fitur-fitur gambar (Tiwari, V. et al., 2020). *Convolution Layer* merupakan kunci untuk mengambil fitur-fitur penting pada gambar dengan menggunakan filter yang akan melakukan operasi perkalian pada setiap nilai piksel yang dilewati oleh filter, fungsi layer ini adalah untuk mengurangi parameter pelatihan (Chen, E. et al., 2019). *Convolution layer* akan menghitung operasi konvolusi dari gambar input menggunakan filter kernel untuk mengekstrak fitur-fitur penting gambar. Filter kernel memiliki dimensi yang sama dengan gambar input tetapi dengan parameter yang lebih kecil daripada ukuran gambar input (Singh, S. A. et al., 2020).

Ukuran kernel 3x3 pada lapisan konvolusi memiliki sejumlah keunggulan. Penggunaan beberapa lapisan 3x3 membuat lapisan konvolusi lebih selektif dan mengurangi jumlah parameter yang harus dipelajari, mengurangi beban komputasi dibandingkan dengan lapisan yang menggunakan kernel 5x5 atau 7x7. Dengan menerapkan dua lapisan 3x3, hasil yang setara dengan lapisan 5x5 dapat dicapai. Demikian pula, dengan menerapkan tiga lapisan 3x3, hasil yang setara dengan lapisan 7x7 dapat dicapai (Simonyan, K., & Zisserman, A., 2014).

Gambar 2.2 merupakan cara kerja *convolution layer*.



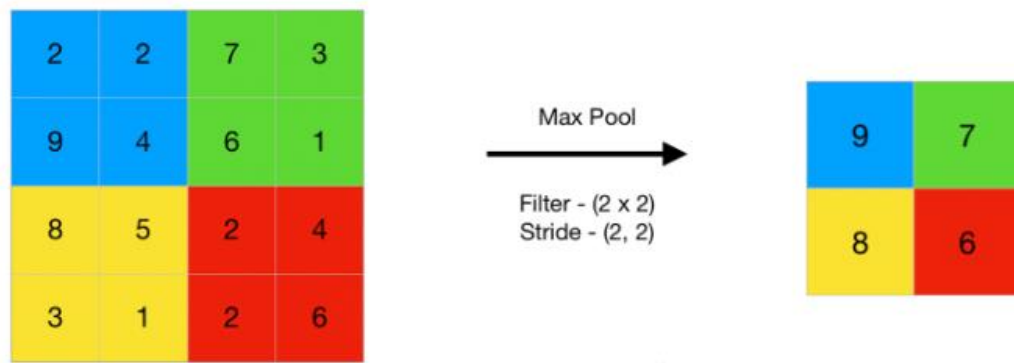
Gambar 2.2 *Convolution layer*

2.1.2.2. Pooling Layer

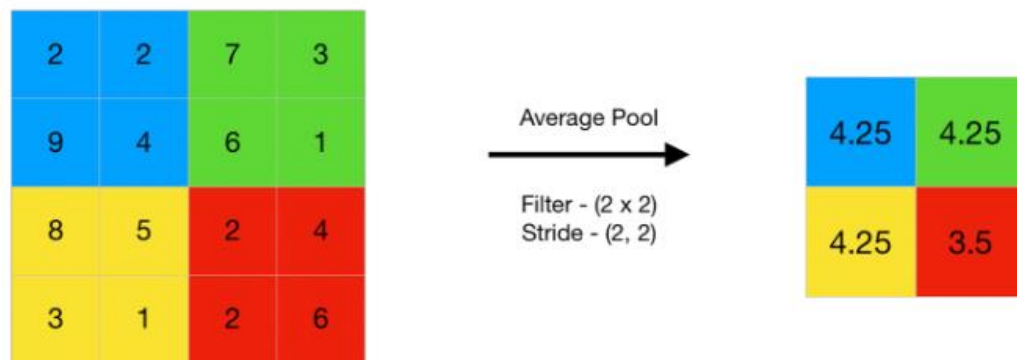
Pooling Layer menggunakan *feature map* yang diambil oleh *convolution layer* untuk memperkecil ukuran *feature map* tersebut dengan *mean pooling* atau *max pooling*, fungsi dari *pooling layer* adalah untuk memperkecil ukuran *feature map* dan mengontrol *over-fitting* (Chen, E. et al., 2019). Fungsi dari *pooling layer* adalah untuk mengurangi ukuran *feature map* yang didapat dari *convolution layer* yang akan mengurangi jumlah parameter pelatihan dan menyederhanakan kompleksitas keseluruhan model (Tiwari, V. et al., 2020). Setelah setiap *convolution layer* terdapat *pooling layer*. *Pooling layer* berfungsi untuk mengurangi dimensi dari *feature map*, setelah mengurangi dimensi pada *feature map* outputnya akan dimasukkan ke *convolution layer* berikutnya yang diperoleh melalui proses *downsampling*, proses ini akan mengakibatkan hilangnya beberapa informasi yang akan membantu mengurangi *over-fitting* dan beban komputasi. *Pooling layer* akan diterapkan ke keseluruhan area dari input secara bertahap dan akan mengubahnya menjadi nilai tertentu berdasarkan metode *subsampling* yang digunakan, metode *subsampling maximum* memiliki performa yang lebih baik dibandingkan metode *subsampling rata-rata* (Singh, S. A. et al., 2020).

Sebaiknya ukuran kernel yang digunakan pada *pooling layer* adalah 2x2 karena jika kita menggunakan kernel 1x1, dimensi tidak akan mengalami perubahan. Sebaliknya, jika memilih kernel dengan ukuran lebih dari 2x2, itu akan mengakibatkan penurunan dimensi dengan tingkat yang signifikan, yang dapat menyebabkan kehilangan informasi penting (Harish, B. S. et al., 2020).

Gambar 2.3 dan gambar 2.4 merupakan cara kerja *Max pooling* dan *average pooling*.



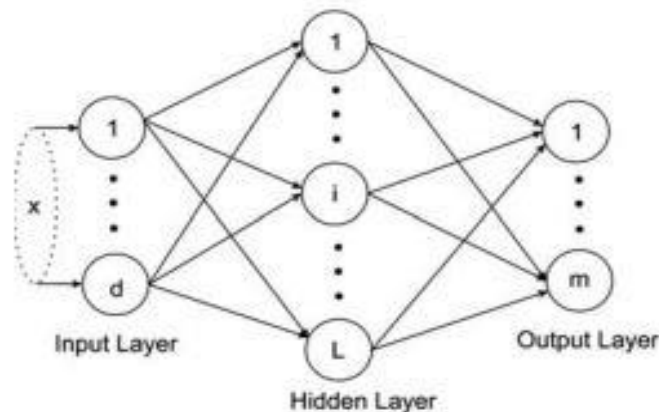
Gambar 2.3 *Max Pooling layer*



Gambar 2.4 *Average Pooling layer*

2.1.3. *Extreme Learning Machine (ELM)*

ELM merupakan metode pembelajaran satu arah yang terdiri atas sebuah layer *input*, sebuah *hidden layer*, dan sebuah layer *output*. Aritektur dari ELM mudah dimengerti dan tidak memerlukan pengaturan parameter secara berulang yang membuat proses pelatihan yang lebih cepat dan mendapatkan performa yang memuaskan dalam klasifikasi (Nahiduzzaman, M. et al., 2021). ELM diajukan sebagai salah satu metode *machine learning* yang berhasil untuk mengklasifikasikan pola dan dianggap sebagai algoritma pembelajaran yang menjanjikan dibandingkan dengan algoritma lain seperti *backward propagation* (BP) *neural network* dan *Support Vector Machine* (SVM) dikarenakan ELM memiliki beberapa keunggulan seperti lebih efisien secara komputasi, mendapatkan hasil yang sama atau lebih baik dibandingkan dengan BP dan SVM dan lain-lain (Samat, A. Et al., 2014). Gambar 2.5 merupakan struktur ELM.



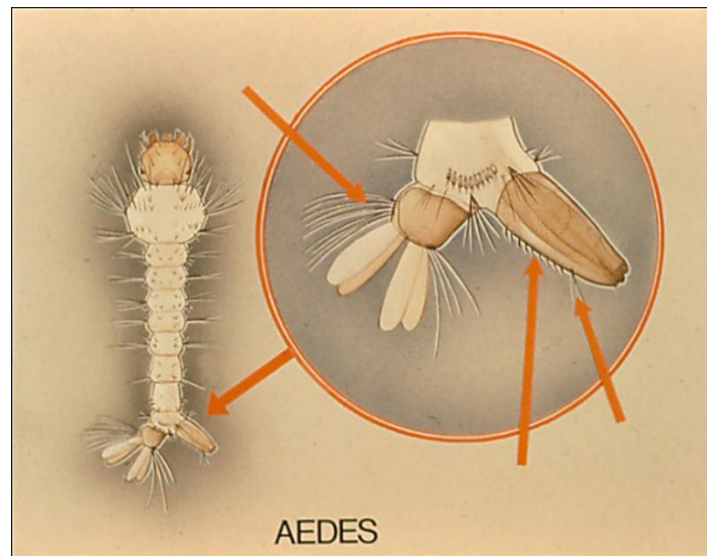
Gambar 2.5 Struktur ELM

2.1.3.1. High-Performance Extreme Learning Machine (HP-ELM)

High-Performance Extreme learning machine (HP-ELM) merupakan *toolbox* atau library Python yang ditulis dalam bahasa Python yang menggunakan library seperti *NumPy* (*numerical Python*) dan *SciPy* (*scientific Python*). HP-ELM dibuat untuk mempermudah pengguna dalam mengimplemetasikan ELM dan agar pengguna dapat mengimplementasikan ELM yang memiliki performa terbaik, ELM dapat mencapai performa terbaik dengan seleksi parameter, regularisasi dan lain-lain (Akusok, A. et al., 2015).

2.2. Nyamuk *Aedes*

Nyamuk memiliki empat tahap dalam daur hidupnya, yaitu tahap pertama adalah telur, tahap kedua adalah larva yang lalu akan menjadi kepompong yang akan menjadi nyamuk dewasa, nyamuk *Aedes* dapat menularkan penyakit seperti demam berdarah dengue, *chikungunya*, virus *zika*, penyakit mayaro, penyakit *yellow fever*, dan *Rift Valley fever* (Sesulihatien, W. T et al 2020). Nyamuk *Aedes* biasa dapat ditemukan di daerah tropis karena kelembaban yang tinggi dikarenakan musim hujan yang sering terjadi, oleh karena itu daerah tropis merupakan daerah yang cocok sebagai tempat berkembang biak nyamuk *Aedes Aegypti* dan *Aedes Albopictus* (Zainol Azman, M. I. A. B., & Sarlan, A. B. 2020). Gambar 2.6 merupakan morfologi larva *Aedes*.



Gambar 2.6 Morfologi Larva *Aedes*

2.3. Klasifikasi Gambar

Klasifikasi gambar adalah masalah yang dasar dan masalah yang masih ada pada *computer vision*, klasifikasi gambar juga banyak digunakan di berbagai bidang seperti klasifikasi penyakit daun tanaman, klasifikasi ekspresi wajah, dan lain-lain. Untuk membuat gambar yang berukuran besar dapat lebih mudah untuk digunakan, digunakanlah *deep neural network* (Tiwari, V. Et al.,2020).

2.4. Optuna

Optuna merupakan software open-source untuk mengoptimasi *hyperparameter* yang bekerja dengan prinsip *define-by-run* yang membuat pengguna dapat mengatur ruang pencarian *hyperparameter* saat program dijalankan. Optuna memiliki algoritma *sampling* yang efisien, berarti optuna dapat mencari *hyperparameter* dengan lebih efisien dan algoritma *pruning* yang dapat ditentukan oleh pengguna, algoritma *pruning* dapat menghentikan penggunaan kombinasi *hyperparameter* yang akan menghasilkan hasil yang buruk berdasarkan kombinasi *hyperparameter* yang telah digunakan sebelumnya. Optuna memiliki arsitektur yang mudah beradaptasi dan dapat digunakan untuk bermacam-macam tugas (Akiba, T. et al., 2019).

2.4.1. *Hyperparameter*

Performa dari model *machine learning* memiliki hubungan erat dengan *training* model tersebut, *training* adalah proses untuk mencari parameter terbaik yang akan memberikan performa terbaik. Namun sebelum memulai proses *training*, diperlukan untuk mengatur beberapa parameter seperti arsitektur model, *learning rate*, dan *optimizer* yang digunakan. Parameter tersebut diketahui sebagai *hyperparameter* yang sudah ditetapkan secara umum oleh ahli machine-learning berdasarkan pengalaman dan intuisi mereka (Lacerda, P. et al., 2021). Parameter adalah nilai yang berhubungan dengan algoritma yang dapat dipelajari melalui pelatihan model, sedangkan *hyperparameter* adalah nilai dari parameter yang mengontrol proses *training* model, *hyperparameter* mengontrol keseluruhan proses *training* dan tidak dapat dipelajari melalui *training* model dan *hyperparameter* tidak dapat diubah ketika model sedang *training* (Shekhar, S. Et al., 2021).

2.5. *Confusion Matrix*

Confusion matrix merupakan salah satu cara metrik evaluasi yang populer digunakan untuk mengetahui kualitas performa yang diberikan oleh model klasifikasi *machine learning* karena *confusion matrix* dapat memberikan gambaran berapa banyak sampel yang diklasifikasi dengan benar dan salah, dan dengan menggunakan *confusion matrix* dapat dihasilkan nilai akurasi, presisi, dan *recall* (Bisong, E., 2019). Nilai yang dihasilkan oleh *confusion matrix* pada masalah klasifikasi dua kelas, ditampilkan dalam bentuk tabel pada gambar 2.7.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

Gambar 2.7 *Confusion Matrix*

Empat nilai yang dihasilkan oleh *confusion matrix* pada gambar 2.6 adalah:

- *True positive (TP)*: *True positive* adalah jumlah sampel yang diprediksi positif dan memiliki kelas positif
- *False positive (FP)*: *False positive* adalah jumlah sampel yang diprediksi positif namun memiliki kelas negatif
- *True negative (TN)*: *True negative* adalah jumlah sampel yang diprediksi negatif dan memiliki kelas negatif
- *False negative (FN)*: *False negative* adalah jumlah sampel yang diprediksi positif namun memiliki kelas negatif

Dari empat nilai tersebut dapat digunakan untuk mendapatkan informasi lain seperti:

- Akurasi: akurasi menghitung hasil sampel yang diprediksi dengan benar yang dihasilkan oleh algoritma pembelajaran. Akurasi mewakili kemampuan model untuk mengidentifikasi kelas dengan benar. Akurasi dihitung dengan menggunakan rumus pada rumus 2.1 (Bisong, E., 2019).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.1)$$

- Presisi: presisi menghitung hasil sampel yang diprediksi dengan benar yang akan dibandingkan dengan seluruh hasil sampel yang memiliki nilai benar atau positif menurut algoritma. Presisi mewakili kemampuan model untuk

mengidentifikasi kelas positif dengan benar. Presisi dihitung dengan rumus pada rumus 2.2 (Bisong, E., 2019).

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

- *Recall*: *recall* menghitung hasil sampel yang diprediksi dengan benar yang akan dibandingkan dengan seluruh hasil sampel yang sesungguhnya bernilai benar atau positif. *Recall* mewakili kemampuan model untuk mengidentifikasi semua kelas positif. *Recall* dihitung dengan rumus pada rumus 2.3 (Bisong, E., 2019).

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

- *F1-score*: *F1-score* merupakan metrik utama yang digunakan untuk menilai performa klasifikasi. *F1-score* mewakili keseluruhan kemampuan model. *F1-score* dapat dihitung dengan rumus pada rumus 2.4 (Pratama, N. H. et al., 2022).

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (2.4)$$

2.6. Activation Functions

Activation functions berfungsi untuk menentukan apakah suatu neuron dalam jaringan saraf akan melanjutkan informasinya ke layer selanjutnya. Neuron yang akan dilanjutkan informasinya ditentukan dengan menjalankan hasil penjumlahan bobot dan bias yang terdapat di dalam neuron melalui fungsi *non-linear* yang akan memutuskan apakah neuron tersebut akan dilanjutkan ke layer selanjutnya atau tidak. *Activation functions* menentukan apakah suatu neuron memiliki informasi yang akan menghasilkan prediksi yang benar pada lapisan output untuk suatu pengamatan dalam dataset pelatihan. *Activation functions* memiliki cara kerja yang mirip dengan cara neuron berkomunikasi dan mentransfer informasi di dalam otak. *Activation functions* juga disebut juga sebagai *non-linearities* karena *activation functions* memberikan kemampuan *non-linear* ke dalam jaringan dan dapat mempelajari pemetaan dari *input* ke *output* untuk *dataset* yang memiliki struktur dasar yang *non-linear* (Bisong, E. 2019). *Activation function* yang tepat dapat membuat model memiliki kemampuan yang lebih baik dalam memetakan data.

Sigmoid functions dan *tanh* banyak digunakan dalam model klasifikasi di awal perkembangan model, namun kedua fungsi tersebut sering menyebabkan *exploding and vanishing gradient*, dengan fungsi ReLU masalah tersebut dapat diatasi. Diantara *activation function* yang umum digunakan, ReLU dianggap yang terbaik (Wang, Y. Et al., 2020).

2.6.1. Sigmoid Function

Sigmoid function adalah fungsi *non-linear* yang mengubah aktivasi agar berada diantara 0 dan 1. Hal ini membuat angka negatif dan positif yang besar menjadi 0 dan 1. Kekurangan dari *sigmoid function* adalah rentan terhadap fenomena yang disebut *exploding and vanishing gradient*. Dalam proses optimasi parameter jaringan saat proses *backpropagation*, nilai gradien dapat menjadi sangat kecil atau besar secara tidak proporsional dengan aktivasi yang terkonsentrasi pada nilai 0 atau 1, ketika hal ini terjadi, dikatakanlah bahwa gradien telah jenuh. Oleh karena itu, perkalian selanjutnya melalui *backpropagation* akan menyebabkan nilai gradien menghilang atau meledak yang akan menyebabkan neuron yang terpengaruh akan menjadi tidak aktif dan tidak dapat mentransfer informasi sehingga mempengaruhi proses pelatihan secara negatif. Kekurangan yang lain adalah output dari *sigmoid function* tidak *zero-centered*, akibat dari ini adalah selama proses *backpropagation*, nilai gradien dapat menjadi seluruhnya positif atau seluruhnya negatif, hal ini akan mempengaruhi proses pelatihan secara negatif (Bisong, E. 2019).

Sigmoid function menggunakan rumus pada rumus 2.5.

$$f(X) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

2.6.2. Hyperbolic Tangent (tanh)

Tanh adalah *sigmoid function* yang ditingkatkan dengan membatasi outputnya menjadi dalam jarak antara -1 dan 1 yang dapat membantu model belajar lebih cepat karena jarak yang lebih besar, peningkatan yang lain adalah output dari tanh sudah *zero-centered*, yang berarti tanh memiliki *mean* yang mendekati 0, ini dapat membantu model belajar lebih cepat dibandingkan dengan *sigmoid function* yang memiliki *mean* sekitar 0.5 yang membuat output cenderung positif. Namun tanh masih mengalami masalah *exploding and vanishing gradient* (Bisong, E. 2019).

Tanh menggunakan rumus pada rumus 2.6.

$$f(X) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.6)$$

2.6.3. Rectified Linear Unit (ReLU)

ReLU bekerja dengan mengatur aktivasi menjadi 0 ketika nilai x kurang dari 0 dan mengatur aktivasi menjadi sesuai dengan nilai x ketika nilai x lebih dari 0. ReLU memiliki kelebihan signifikan dibandingkan dengan *tanh function* dan *sigmoid function* dengan memberikan solusi parsial untuk masalah *exploding and vanishing gradient*. Namun beberapa nilai gradien masih dapat mengalami *exploding* dan *vanishing* saat proses *backpropagation* dengan *learning rate* yang besar, tetapi dengan *learning rate* yang telah ditentukan dengan baik, masalah tersebut dapat dihindari (Bisong, E. 2019). ReLU menggunakan rumus pada rumus 2.7.

$$f(X) = \max(0, x) \quad (2.7)$$

2.7. Normalization

Normalization adalah teknik *scaling* atau teknik pemetaan atau tahap pra-proses. Dengan *normalization*, nilai piksel gambar dapat dibatasi agar dapat menghilangkan perbedaan nilai piksel yang terlalu jauh, ini sangat membantu dalam melakukan prediksi, dan membuat nilai piksel gambar lebih konsisten. Ada banyak metode untuk melakukan prediksi dengan hasil yang sangat bervariasi, *normalization* berfungsi untuk membuat hasil-hasil tersebut lebih serupa (Patro, S. G. K., & Sahu, K. K., 2015).

2.8. Luminosity

Metode *luminosity* adalah metode yang lebih mutakhir dari metode *averaging* dan metode *lightness*. Metode ini juga meratakan nilai gambar, tetapi metode *luminosity* membentuk rata-rata yang berbobot sesuai dengan persepsi manusia. Manusia lebih sensitif terhadap warna hijau dibandingkan dengan warna lain, oleh karena itu warna hijau diberi bobot yang paling berat. Metode *luminosity* menggunakan rumus pada rumus 2.8 di mana R adalah *Red*, G adalah *Green*, dan B adalah *Blue* (Reddy, R. V. K. Et al., 2016).

$$Luminosity = 0.21R + 0.72G + 0.07B \quad (2.8)$$

BAB III

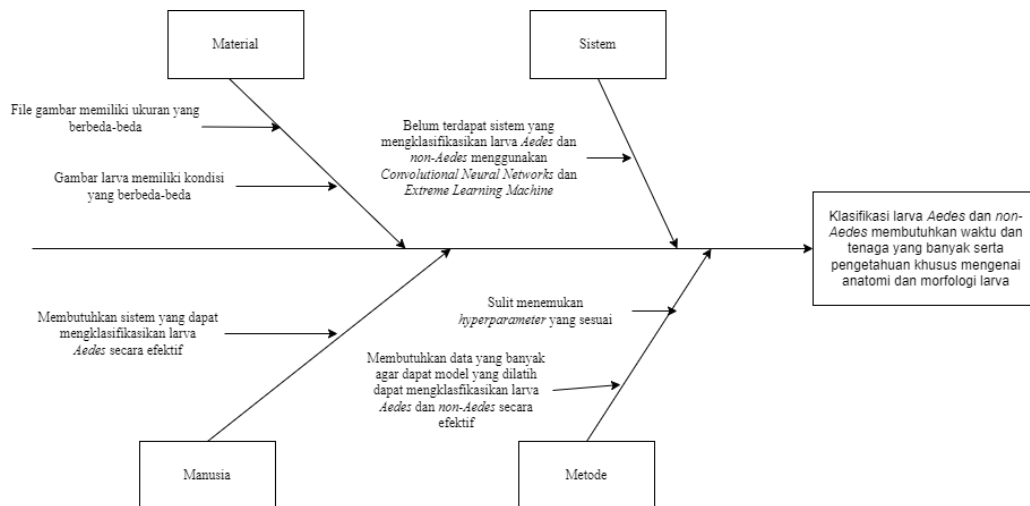
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem merupakan tahap awal dalam pembuatan sistem yang bertujuan untuk mendapatkan solusi untuk masalah pada sistem dengan cara membedah masalah menjadi bagian-bagian agar dapat dipelajari dan dievaluasi. Analisis sistem terdiri atas analisis masalah dan analisis kebutuhan.

3.1.1. Analisis masalah

Analisis masalah merupakan sebuah proses yang bertujuan untuk mengidentifikasi masalah yang dihadapi oleh sistem agar sistem dapat mengatasi masalah tersebut dan berjalan sesuai dengan tujuannya. Permasalahan inti pada penelitian ini adalah sistem yang dapat mengklasifikasikan larva *Aedes* dan *non-Aedes* secara efektif. Analisis masalah pada penelitian ini dilakukan dengan menggunakan diagram Ishikawa (*fishbone diagram*), diagram Ishikawa (*fishbone diagram*) berfungsi untuk mengidentifikasi sebab-akibat dari sebuah masalah. Gambar 3.1 merupakan Diagram Ishikawa (*fishbone diagram*) penelitian ini.



Gambar 3.1 Diagram Ishikawa (*Fishbone Diagram*)

Menurut gambar 3.1, permasalahan inti pada penelitian ini adalah klasifikasi larva *Aedes* dan *non-Aedes* membutuhkan waktu dan tenaga yang banyak serta pengetahuan khusus mengenai anatomi dan morfologi larva. Oleh karena itu, solusi untuk permasalahan tersebut adalah dengan membangun sistem yang dapat

mengklasifikasikan larva *Aedes* dan *non-Aedes* menggunakan *Convolutional Neural Networks* dan *Extreme Learning Machine*.

3.1.2. Analisis kebutuhan

Analisis kebutuhan merupakan sebuah proses yang bertujuan untuk mendapatkan dan menjelaskan kemampuan dari sistem yang akan dibuat. Analisis kebutuhan terdiri atas kebutuhan fungsional dan non-fungsional.

- Kebutuhan fungsional

Kebutuhan fungsional adalah penjelasan tentang fitur-fitur yang dapat dilakukan oleh sistem. Kebutuhan fungsional pada sistem ini adalah:

1. Jenis file yang dapat diterima oleh sistem adalah *.JPEG dan *.PNG
2. Sistem dapat mengekstraksi fitur gambar menggunakan *Convolutional Neural Networks*.
3. Menggunakan fitur gambar yang telah diekstraksi oleh *Convolutional Neural Networks*, sistem dapat mengklasifikasi gambar larva *Aedes* dan *non-Aedes* menggunakan *Extreme Learning Machine*.
4. Sistem dapat melakukan pra-proses pada gambar dengan mengubah warna gambar menjadi *grayscale* dan melakukan normalisasi dengan membagi piksel gambar dengan 255.
5. Sistem dapat menampilkan hasil klasifikasi kepada pengguna.

- Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah penjelasan tentang bagaimana sistem berfungsi, tanpa secara langsung berkaitan dengan fitur-fitur sistem, seperti *user interface*, keamanan, dan lain-lain. Kebutuhan non-fungsional pada sistem ini adalah:

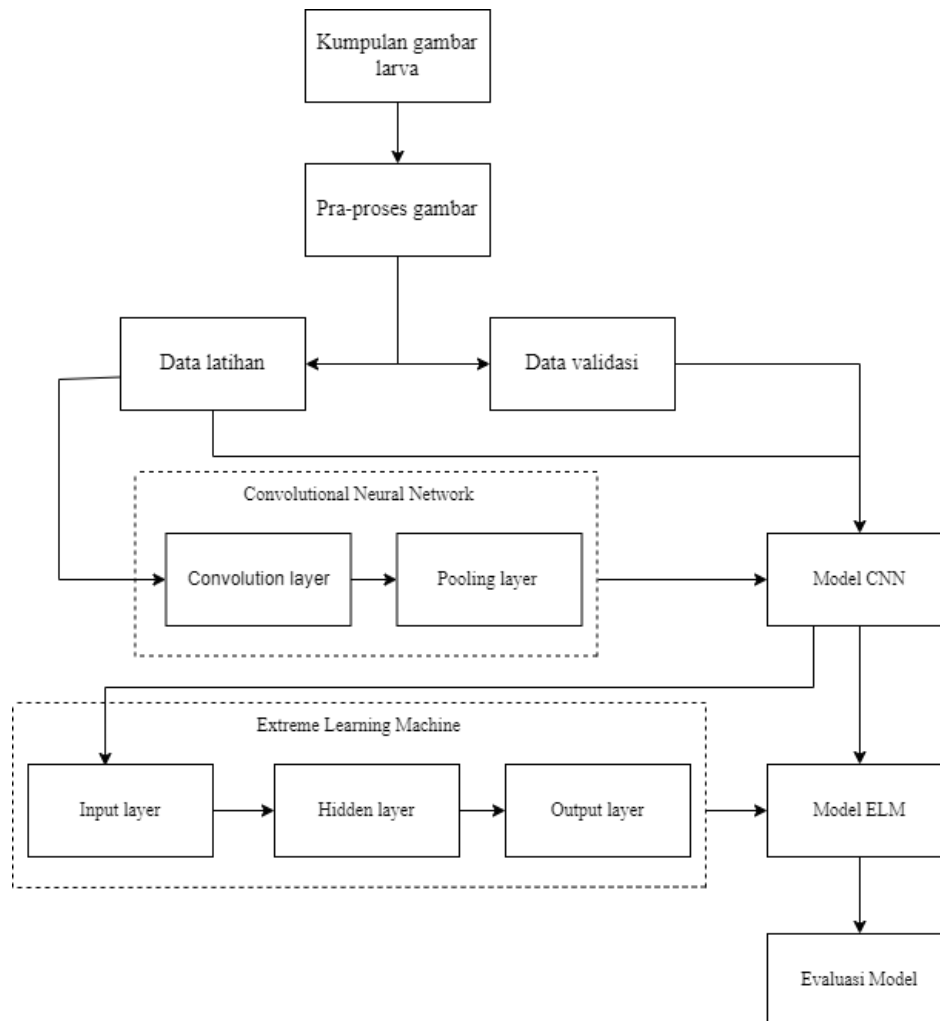
1. *User friendly*
Sistem memiliki *user interface* yang mudah dimengerti oleh pengguna.
2. Performa
Sistem dapat mengklasifikasikan gambar larva *Aedes* dan *non-Aedes* serta menampilkan hasil klasifikasi kepada pengguna.
3. Akurasi
Sistem dapat mengklasifikasikan gambar larva *Aedes* dan *non-Aedes* dengan akurasi yang tinggi.

3.2. Penjelasan Penelitian

Pada penelitian ini, dikembangkanlah sistem untuk mengklasifikasikan jenis larva nyamuk *Aedes* dan non-*Aedes*. Sistem dikembangkan dengan menggunakan *Convolutional Neural Network* (CNN) untuk mengekstraksi fitur pada gambar, dan *Extreme Learning Machine* (ELM) untuk mengklasifikasikan jenis larva. Model CNN-ELM akan dilatih dengan gambar yang telah di pra-proses menggunakan metode *resize* gambar dan normalisasi gambar.

Penelitian ini dimulai dengan mengumpulkan data dari Balai Teknik Kesehatan Lingkungan Dan Pengendalian Penyakit (BTKLPP) Kelas I Medan dengan menggunakan mikroskop digital, kemudian dataset akan disimpan di google drive dan diakses menggunakan google colaboratory. Dengan dataset tersebut, tahap pra-proses dapat dimulai dengan menggunakan metode *resize* setelah dataset di *resize*, dataset akan diubah menjadi array numerik supaya gambar dapat dinormalisasi. Lalu dataset akan dibagi menjadi dataset latihan dan dataset validasi, lalu dengan menggunakan dataset yang telah melalui pra-proses tahap pelatihan model CNN-ELM dimulai dengan model CNN menerima dataset yang akan di proses oleh *Convolution layer* CNN. Proses dimulai dengan *Convolution layer* membagi gambar menjadi beberapa bagian menggunakan filter kernel yang akan melewati seluruh bagian gambar, proses ini akan mengambil fitur-fitur penting gambar. Kemudian *pooling layer* CNN akan mengurangi ukuran *feature map* dengan menggunakan *mean pooling* atau *max pooling*, efek dari proses ini adalah over-fitting dapat dikontrol, mengurangi jumlah parameter pelatihan dan menyederhanakan kompleksitas keseluruhan model. Pada model ELM, *input layer* ELM akan menerima fitur penting gambar yang telah di ambil oleh CNN dan meneruskannya ke *hidden layer* yang akan melakukan perhitungan berdasarkan *hidden neuron* yang telah ditetapkan, output dari *hidden layer* akan diteruskan ke *output layer* yang akan mengklasikasikan jenis larva nyamuk, model ELM akan dilatih dengan struktur tersebut dengan menggunakan *neuron* dan *activation function* yang sudah ditetapkan. Evaluasi model ini akan dilakukan dengan menggunakan *confusion matrix* yang akan memberikan nilai *true positive*, *true negative*, *false positive*, *false negative*, *precision*, *recall*, *f1-score*, dan *accuracy*,

nilai-nilai tersebut dapat digunakan untuk mengevaluasi sebaik apa model bekerja. Alur penelitian yang menggambarkan alur penelitian dapat dilihat pada gambar 3.2.



Gambar 3.2 Alur Penelitian

3.2.1. Sumber Dataset

Dataset yang digunakan didapat dari Balai Teknik Kesehatan Lingkungan Dan Pengendalian Penyakit (BTKLPP) Kelas I Medan. Dataset merupakan larva nyamuk yang diambil dengan menggunakan mikroskop digital, ukuran minimal piksel gambar dataset yang akan digunakan untuk melatih model adalah 640 x 480 piksel agar gambar dapat di pra-proses oleh model CNN-ELM.

3.2.2. Pra-proses gambar

Selanjutnya gambar-gambar pada dataset di pra-proses dengan mengubah gambar menjadi array numerik dengan menggunakan *img_to_array* dari library tensorflow

kemudian array numerik dinormalisasikan dengan membagi array numerik dengan 255 agar nilai piksel bernilai 0 atau 1 yang akan memudahkan model dalam menganalisis gambar, dan meningkatkan kontras gambar yang akan meningkatkan akurasi hasil klasifikasi gambar. Data yang telah di pra-proses akan dibagi menjadi data *training* dan data *testing*.

3.2.3. Ekstraksi Fitur dengan CNN

Selanjutnya gambar yang telah di pra-proses akan diambil fitur-fitur pentingnya dengan menggunakan CNN, fitur-fitur penting gambar diekstrak oleh *convolution layer* dan *pooling layer* akan memperkecil dimensi gambar, lalu *flatten layer* akan mengubah array numerik yang telah diproses oleh seluruh *convolution layer* dan *pooling layer* yang berbentuk vektor multidimensi menjadi vektor satu dimensi agar dapat diterima oleh ELM, lalu model CNN akan dilatih agar *convolution layer* dapat mengambil fitur-fitur penting gambar dengan lebih baik.

3.2.4. Klasifikasi dengan ELM

Selanjutnya gambar yang telah diproses oleh CNN akan dimasukkan ke model ELM, input layer ELM akan mengambil output dari *flatten layer* model CNN, setelah itu hidden layer akan melakukan perhitungan berdasarkan *hyperparameter* yang sudah ditetapkan sebelum proses latihan dimulai. *Hyperparameter* yang akan digunakan adalah banyaknya neuron, dan jenis *activation function* yang akan digunakan oleh model ELM. Diakhiri dengan output layer yang akan menentukan jenis larva nyamuk, dengan struktur tersebut model ELM akan dilatih agar dapat menentukan jenis larva nyamuk.

3.2.5. Evaluasi model

Selanjutnya model CNN-ELM akan dievaluasi menggunakan confusion matrix yang akan menghasilkan nilai *true positive*, *true negative*, *false positive*, dan *false negative*, lalu dari nilai-nilai tersebut dapat dihitung nilai akurasi, presisi, *recall*, dan *f1-score* dengan menggunakan rumus yang terdapat pada gambar 2.7, 2.8 dan 2.10. Dengan nilai-nilai tersebut dapat diketahui sebaik apa model CNN bekerja.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Perangkat yang Digunakan

Perangkat yang digunakan untuk membuat model CNN-ELM untuk mengklasifikasikan larva *Aedes* adalah sebagai berikut:

4.1.1. Perangkat Keras

Detail perangkat keras yang digunakan untuk membuat sistem adalah sebagai berikut:

1. Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz
2. 16.0 GB RAM
3. Intel® HD GRAPHICS 4600
4. NVIDIA GeForce GTX 765M

4.1.2. Perangkat Lunak

Detail perangkat lunak yang digunakan untuk membuat sistem adalah sebagai berikut:

1. Google Colaboratory
2. Google Drive
3. Sistem Operasi Windows 10 Enterprise 64-bit

4.1.3. Library

Detail library yang digunakan untuk membuat sistem adalah sebagai berikut:

1. Optuna
2. Os
3. Hpelm
4. Tensorflow
5. Numpy
6. Keras
7. Sklearn
8. Pickle

4.2. Penerapan Tahap Pengumpulan Dataset

Dataset yang digunakan untuk melatih dan mengevaluasi model didapat dari Balai Teknik Kesehatan Lingkungan Dan Pengendalian Penyakit (BTKLPP) Kelas I Medan. Dataset dikumpulkan menggunakan mikroskop digital dan memiliki ukuran piksel sebesar 640 x 480 piksel. Jumlah data larva *Aedes* adalah 1152 gambar dan jumlah data larva *non-Aedes* adalah 74 gambar. Dengan menggunakan *image data generator* dari library tensorflow, tiap gambar akan dimodifikasi agar model tidak mengalami *overfitting*. Jumlah data *Aedes* yang telah dimodifikasi menjadi 2098 gambar, dan jumlah data *non-Aedes* yang telah dimodifikasi menjadi 809 gambar. Gambar 4.1 merupakan contoh gambar dataset yang digunakan.



Gambar 4.1 Contoh Gambar Larva, *Aedes* (atas), *non-Aedes* (bawah)

4.3. Penerapan Tahap Pra-proses Dataset

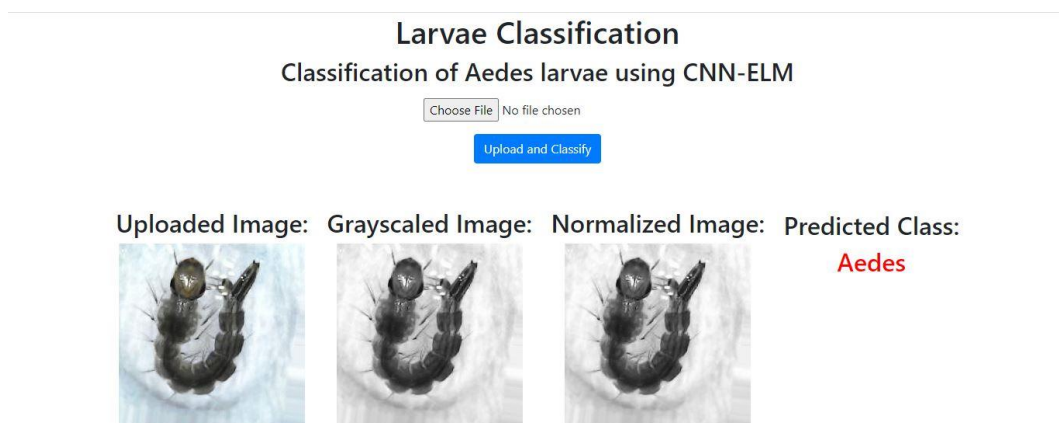
Dataset yang telah dikumpulkan diakses menggunakan google colaboratory dan di *resize* ukuran nya menjadi 240 x 240 piksel agar semua gambar memiliki ukuran yang seragam dan warna gambar diubah dengan menjadi *grayscale* menggunakan metode luminosity, lalu gambar diubah menjadi array numerik dengan *img_to_array* dari library tensorflow, setelah itu array numerik dibagi dengan 255 untuk menormalisasikan gambar agar nilai-nilai piksel gambar terbatas antara 0 dan 1, selanjutnya array numerik yang sudah dinormalisasi ditambah dimensi nya dengan *np.expand_dims* dari library numpy, fungsi dari penambahan dimensi adalah agar array numerik dapat diterima oleh input layer CNN. Dataset yang telah ditambah dimensinya akan dibagi menjadi data *training* dan data validasi.

Gambar 4.2 merupakan kode yang digunakan untuk mengambil data, pra-proses gambar, dan membagi gambar menjadi data latihan dan data validasi. *Load_img* digunakan untuk mengambil gambar dan mengubah warna dan ukuran gambar, lalu gambar diubah menjadi array numerik dengan menggunakan *img_to_array*, Array numerik dibagi dengan 255 untuk dinormalisasi. Setelah data selesai di pra-proses, data akan dibagi menjadi data latihan dan data validasi dengan menggunakan *train_test_split*. Gambar 4.3 sampai gambar 4.12 merupakan contoh gambar yang telah di pra-proses.

```
dataset = []
label = 0
for file in os.listdir(data_dir):
    path = os.path.join(data_dir, file)
    for img in os.listdir(path):
        image = load_img(os.path.join(path, img), color_mode = "grayscale", target_size=(240,240))
        image = img_to_array(image)
        image = image/255.0
        dataset += [[image, label]]
    label+=1
data, labels = zip(*dataset)
data = np.array(data)
labels = np.array(to_categorical(labels))
print('There are', len(dataset), 'images in', label, 'classes')

train_data, val_data, train_labels, val_labels = train_test_split(data, labels, test_size=0.2, random_state=44)
```

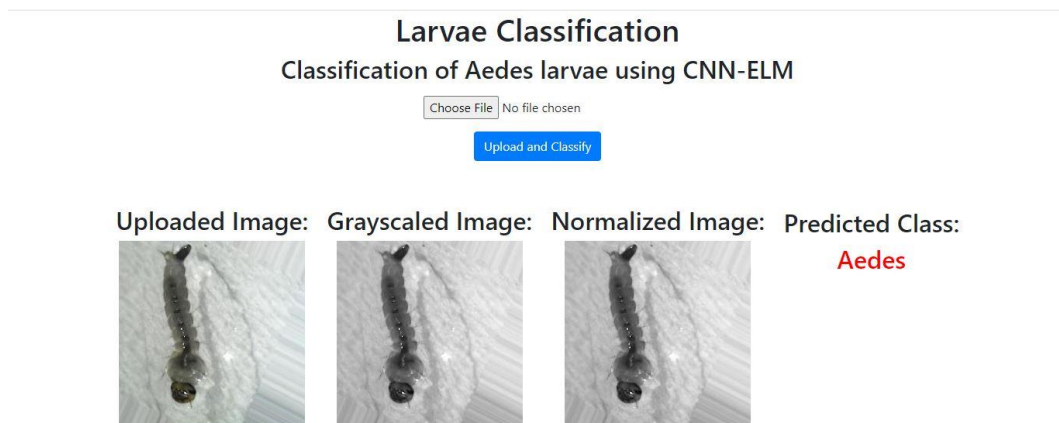
Gambar 4.2 Gambar Kode tahap Pra-proses



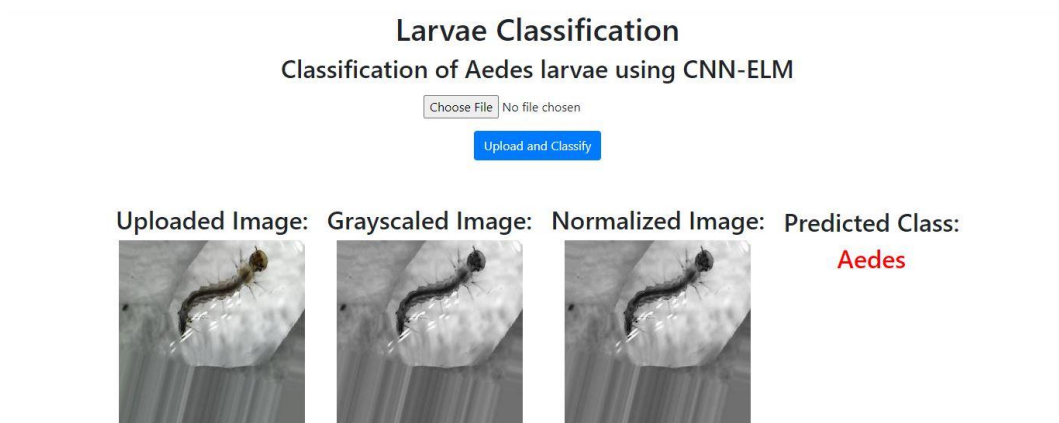
Gambar 4.3 Gambar Larva *Aedes* yang di Pra-proses



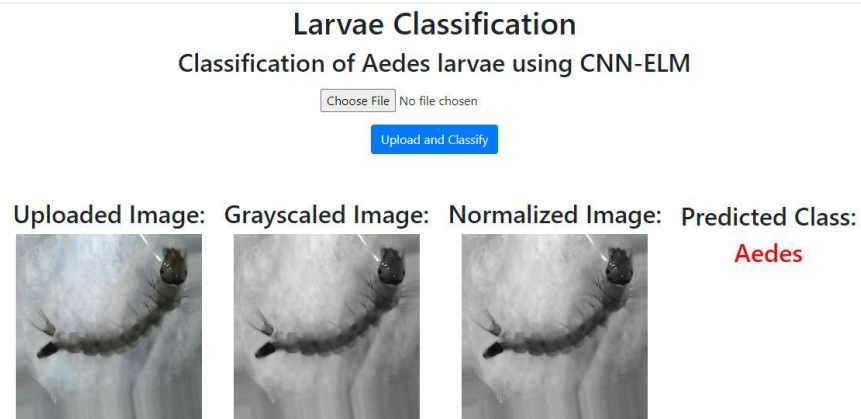
Gambar 4.4 Gambar Larva *Aedes* yang di Pra-proses



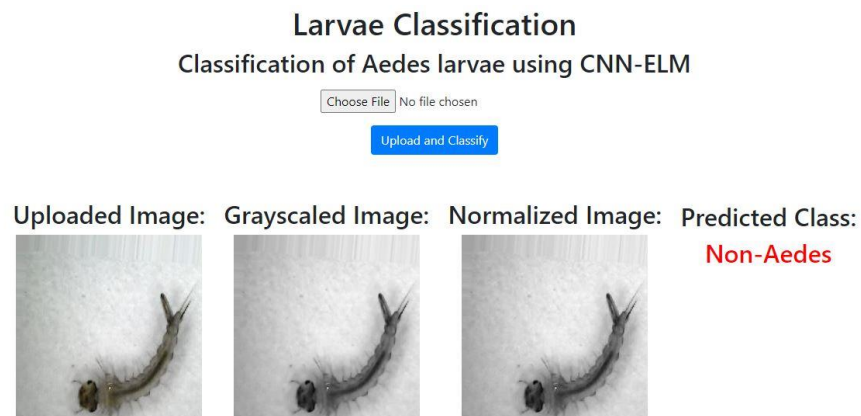
Gambar 4.5 Gambar Larva *Aedes* yang di Pra-proses



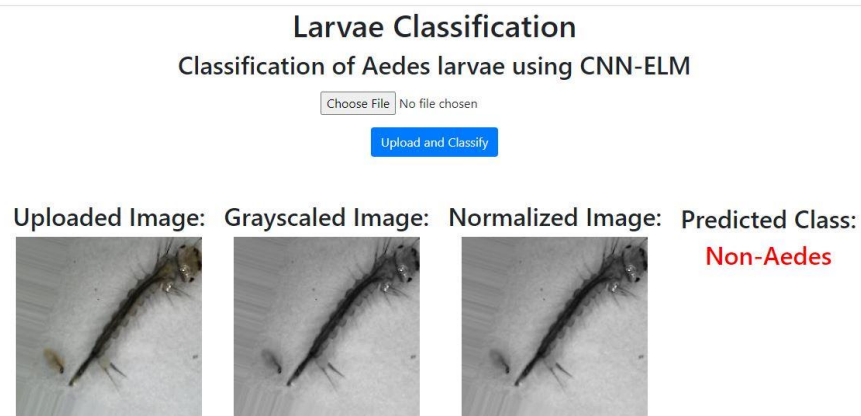
Gambar 4.6 Gambar Larva *Aedes* yang di Pra-proses



Gambar 4.7 Gambar Larva *Aedes* yang di Pra-proses



Gambar 4.8 Gambar Larva *Non-Aedes* yang di Pra-proses



Gambar 4.9 Gambar Larva *Non-Aedes* yang di Pra-proses

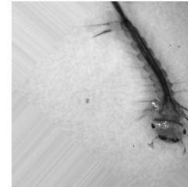
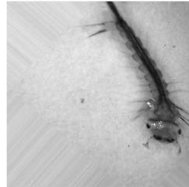
Larvae Classification

Classification of Aedes larvae using CNN-ELM

Choose File No file chosen

Upload and Classify

Uploaded Image: Grayscaled Image: Normalized Image: Predicted Class:



Non-Aedes

Gambar 4.10 Gambar Larva *Non-Aedes* yang di Pra-proses

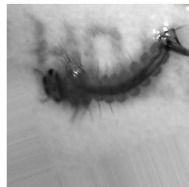
Larvae Classification

Classification of Aedes larvae using CNN-ELM

Choose File No file chosen

Upload and Classify

Uploaded Image: Grayscaled Image: Normalized Image: Predicted Class:



Non-Aedes

Gambar 4.11 Gambar Larva *Non-Aedes* yang di Pra-proses

Larvae Classification

Classification of Aedes larvae using CNN-ELM

Choose File No file chosen

Upload and Classify

Uploaded Image: Grayscaled Image: Normalized Image: Predicted Class:



Non-Aedes

Gambar 4.12 Gambar Larva *Non-Aedes* yang di Pra-proses

4.4. Penerapan Tahap Ekstraksi Fitur dengan CNN

Setelah gambar di pra-proses, model CNN dibuat dengan library keras menggunakan google colaboratory, model CNN yang dibuat memiliki lima pasang *convolution layer* dan *pooling layer*, sebuah *flatten layer*, sebuah *dense layer*, *optimizer Adam* dengan *learning rate* $1e-3$. Pada *convolution layer* jumlah filter yang digunakan adalah 16, 32, 64, 128, dan 256 dengan ukuran filter kernel 3×3 . *Activation functions* yang digunakan pada *convolution layer* berupa ReLU yang dapat membantu model mempelajari hubungan antara fitur-fitur gambar dan mengatasi masalah *exploding and vanishing gradient*, *pooling layer* menggunakan metode *max pooling* karena metode *max pooling* lebih efisien secara komputasi, dapat mengatasi *overfitting*, dan dapat mempertahankan informasi yang paling penting. *Pooling layer* menggunakan ukuran filter kernel 2×2 . Gambar yang telah di pra-proses akan dimasukkan ke *convolution layer* untuk mengekstraksi fitur dan membuat gambar menjadi *feature map* lalu *pooling layer* akan mengurangi dimensi *feature map* dengan *max pooling*. Setelah gambar telah melewati semua *convolution layer* dan *pooling layer*, gambar akan direpresentasikan dengan matriks multidimensi, *flatten layer* akan mengubah matriks multidimensi tersebut menjadi matriks satu dimensi agar *input layer* ELM dapat menerima input tersebut. Setelah itu model CNN akan dilatih agar model CNN dapat mengekstraksi fitur dengan baik. Model CNN akan disimpan dengan ekstensi h5.

Pada gambar 4.13 merupakan kode yang digunakan untuk membuat model cnn, *Conv2D* merupakan *convolution layer* pada layer pertama terdapat *input_shape* yang berfungsi untuk menentukan ukuran yang akan diterima model, *MaxPool2D* merupakan *pooling layer*, *Flatten* merupakan *flatten layer*, *cnn.fit* berfungsi untuk melatih model CNN. Gambar 4.14 sampai gambar 4.17 merupakan contoh fitur yang diekstraksi oleh CNN.


```

cnn = Sequential()
cnn.add(Conv2D(16, (3, 3), activation='relu', input_shape=(train_data.shape[1], train_data.shape[2], 1)))
cnn.add(MaxPool2D(pool_size=(2, 2)))
cnn.add(Conv2D(32, (3, 3), activation='relu'))
cnn.add(MaxPool2D(pool_size=(2, 2)))
cnn.add(Conv2D(64, (3, 3), activation='relu'))
cnn.add(MaxPool2D(pool_size=(2, 2)))
cnn.add(Conv2D(128, (3, 3), activation='relu'))
cnn.add(MaxPool2D(pool_size=(2, 2)))
cnn.add(Conv2D(256, (3, 3), activation='relu'))
cnn.add(MaxPool2D(pool_size=(2, 2)))
cnn.add(Flatten())
cnn.add(Dense(2, activation='sigmoid'))

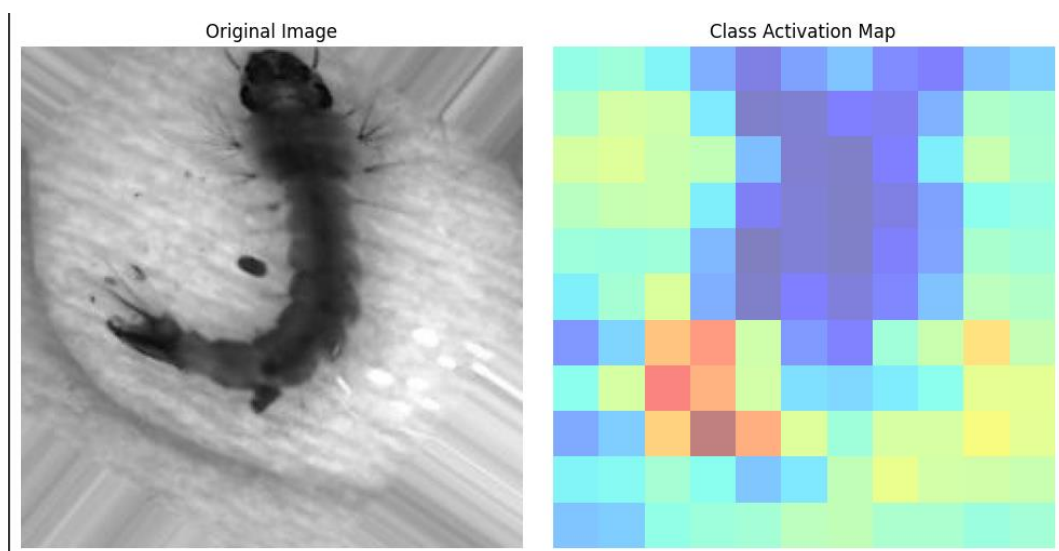
adam = Adam(learning_rate=1e-3)

cnn.compile(optimizer=adam, loss='binary_crossentropy', metrics=['accuracy'])

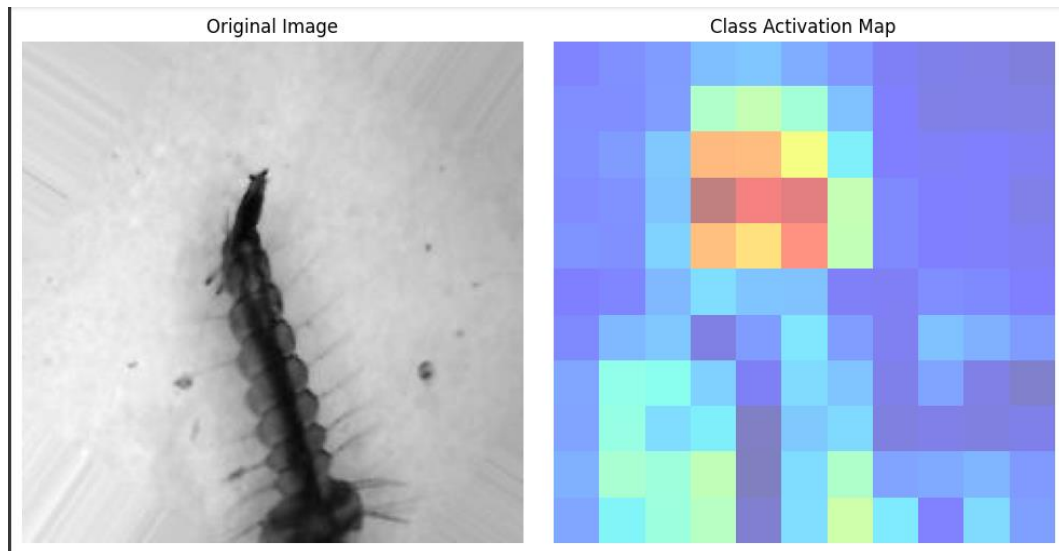
cnn.summary()
cnn_model = cnn.fit(train_datagen.flow(train_data, train_labels, batch_size=50),
                    validation_data=(val_data, val_labels),
                    epochs=20)

```

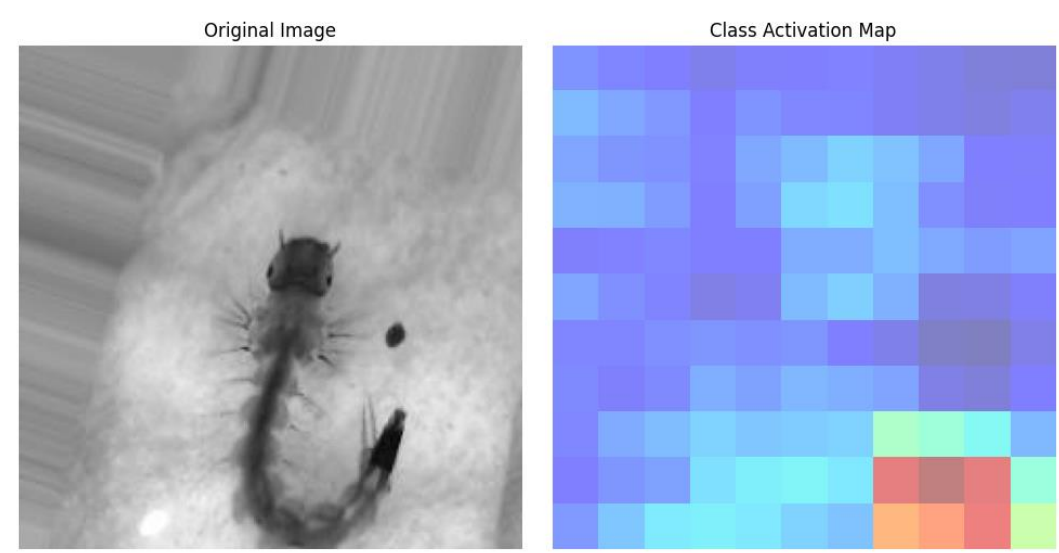
Gambar 4.13 Gambar Kode Model CNN



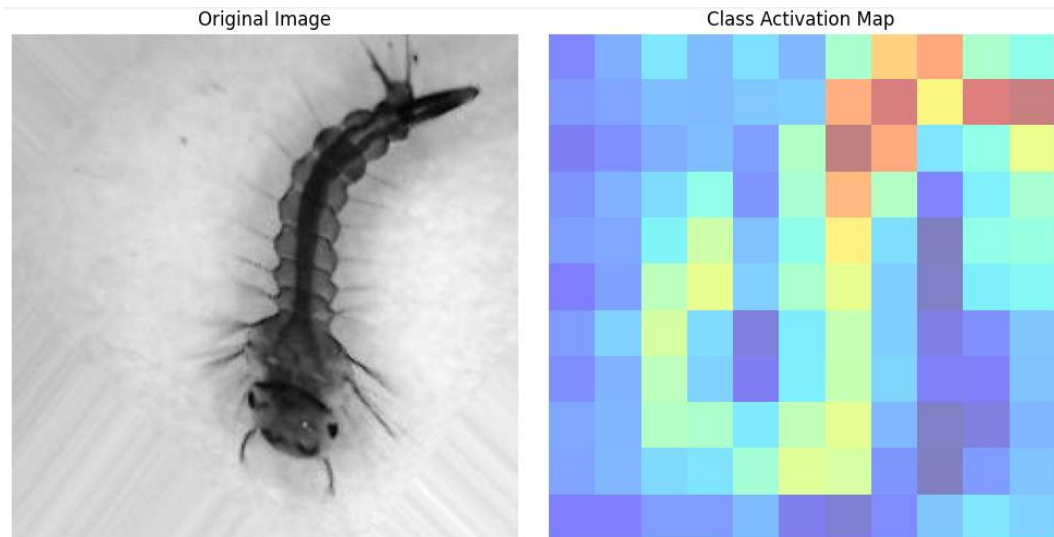
Gambar 4.14 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar *Aedes* (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)



Gambar 4.15 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar *Non-Aedes* (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)



Gambar 4.16 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar *Aedes* (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)



Gambar 4.17 Contoh Fitur yang Mempengaruhi Output CNN pada Gambar *Non-Aedes* (Semakin Merah Warnanya Maka Semakin Berpengaruh Fitur Tersebut)

4.5. Penerapan Tahap Klasifikasi dengan ELM

Setelah fitur gambar telah diekstrak oleh CNN, model ELM dibuat dengan library *hpelm* dan *hyperparameter* terbaik dicari menggunakan *optuna* menggunakan *google colaboratory*, *hyperparameter* ELM yang dapat dicari adalah jumlah neuron dengan rentang 50 sampai 1000 dengan *increment* 50, dan *activation functions* berupa *sigmoid function* dan *tanh*. Akurasi model dicari dengan menggunakan *accuracy score* dari library *sklearn*. *Hyperparameter* akan dicari sebanyak lima kali. Berikut adalah *hyperparameter* yang dihasilkan oleh *Optuna* menggunakan CNN yang belum dilatih dapat dilihat di tabel Tabel 4.1.

Tabel 4.1 *Hyperparameter* diperoleh Optuna dengan CNN belum dilatih

No.	Jumlah Neuron	<i>Activation functions</i>	Akurasi
1.	350	tanh	0,83
2.	400	tanh	0,83
3.	350	sigm	0,84
4.	400	sigm	0,84
5.	500	tanh	0,84

hyperparameter yang dihasilkan oleh Optuna menggunakan CNN yang sudah dilatih dapat dilihat di Tabel 4.2.

Tabel 4.2 *Hyperparameter* diperoleh Optuna dengan CNN sudah dilatih

No.	Jumlah Neuron	<i>Activation functions</i>	Akurasi
1.	350	tanh	0,97
2.	400	sigm	0,97
3.	450	tanh	0,98
4.	650	tanh	0,98
5.	800	sigm	0,98

Hyperparameter yang dihasilkan oleh Optuna yang digunakan adalah:

- Jumlah neuron : 450
- *Activation function* : tanh
- Akurasi : 0,98

Menggunakan *hyperparameter* tersebut, model ELM dilatih dan model ELM disimpan menggunakan ekstensi pickle.

Gambar 4.18 merupakan kode optuna yang digunakan untuk mencari *hyperparameter* untuk model ELM. Dimulai dengan menentukan *hyperparameter* yang akan dicari, yaitu jumlah neuron dengan rentang 50 sampai 1000 dengan *increment* 50, dan *activation functions* berupa *sigmoid function* dan *tanh*, lalu ELM akan dilatih dengan menggunakan *hyperparameter* yang ditentukan oleh optuna, optuna akan mencari *hyperparameter* sebanyak 20 kali dengan menggunakan *study.optimize(objective, n_trials=20)* yang akan mencari *hyperparameter* dengan tujuan untuk memaksimalkan akurasi dengan menggunakan *direction="maximize"*. Gambar 4.19 merupakan kode yang digunakan pada model ELM. *Feature_extractor* berfungsi untuk mengambil *feature map* gambar menggunakan model CNN. Pada *ELM(train_data_features.shape[1], 2)*, *train_data_features.shape[1]* berfungsi sebagai *input layer* yang menerima bentuk *feature map*, dan 2 berfungsi sebagai *output layer* yang menentukan jumlah output yang dihasilkan. *Elm.add_neurons* berfungsi untuk menentukan jumlah neuron pada *hidden layer* ELM dan jenis *activations function* yang digunakan ELM. *Elm.train* berfungsi untuk melatih model ELM.

```

def objective(trial):
    neurons = trial.suggest_int("neurons", 50, 1000, 50)
    activation = trial.suggest_categorical("activation", ['sigm', 'tanh'])

    elm = ELM(X_train.shape[1], 2)
    elm.add_neurons(neurons, activation)
    elm.train(X_train, train_labels, 'c')

    predicted = elm.predict(X_test)

    predicted = np.argmax(predicted, axis=1)

    true_labels = np.argmax(val_labels, axis=1)

    elm_accuracy = accuracy_score(true_labels, predicted)

    return elm_accuracy

study = optuna.create_study(direction="maximize")
study.optimize(objective, n_trials=20)

print("Best number of neurons:", study.best_params['neurons'])
print("Best activation:", study.best_params['activation'])
print("Best ELM accuracy:", study.best_value)

```

Gambar 4.18 Gambar Kode Optuna

```

flatten_name = cnn.layers[-2].name
feature_extractor = Model(inputs=cnn.input, outputs=cnn.get_layer(flatten_name).output)
print(flatten_name)
train_data_features = feature_extractor.predict(train_data)
val_data_features = feature_extractor.predict(val_data)
elm = ELM(train_data_features.shape[1], 2)
elm.add_neurons(450, "tanh")
elm.train(train_data_features, train_labels, "c")

predicted = elm.predict(val_data_features)

predicted_labels = np.argmax(predicted, axis=1)
true_labels = np.argmax(val_labels, axis=1)

elm_accuracy = accuracy_score(true_labels, predicted_labels)

print("ELM accuracy:", elm_accuracy)

```

Gambar 4.19 Gambar Kode Model ELM

4.6. Penerapan Tahap Evaluasi model

Setelah model CNN dan ELM telah dilatih dan disimpan lalu pada google colaboratory, model CNN dan ELM akan dievaluasi menggunakan *confusion matrix* dari library sklearn, hasil evaluasi *confusion matrix* adalah sebagai berikut:

- *True positive* : 424
- *False positive* : 5
- *True negative* : 146
- *False negative* : 7
- Presisi *Aedes* : 0,98
- Presisi *non-Aedes* : 0,97
- *Recall Aedes* : 0,99
- *Recall non-Aedes* : 0,95
- *F1-score Aedes* : 0,99
- *F1-score non-Aedes* : 0,96
- Akurasi : 0,98

Hasil presisi didapat dengan menggunakan rumus pada rumus 2.2, hasil *recall* didapat dengan menggunakan rumus pada rumus 2.3, hasil *F1-score* didapat dengan menggunakan rumus pada rumus 2.4, dan akurasi didapat dengan menggunakan rumus pada rumus 2.1.

Gambar 4.19 merupakan kode yang digunakan untuk mengevaluasi model CNN-ELM, *confusion matrix* digunakan untuk mencari *true positive*, *false positive*, *true negative*, dan *false negative*, dan *classification_report* digunakan untuk mencari presisi, *recall*, *F1-score*, dan akurasi.

```
predicted = elm_model.predict(val_data_features)
predicted_labels = np.argmax(predicted, axis=1)
true_labels = np.argmax(val_labels, axis=1)

accuracy = np.sum(predicted_labels == true_labels) / true_labels.shape[0]
print(f"Accuracy: {accuracy}")

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

cm = confusion_matrix(true_labels, predicted_labels)
print("Confusion matrix:")
print(cm)
print("Classification Report:")
print(classification_report(true_labels, predicted_labels))
```

Gambar 4.20 Gambar Kode Evaluasi Mode

BAB V

KESIMPULAN DAN SARAN

Bab ini akan membahas tentang kesimpulan dan saran dari sistem klasifikasi larva *Aedes* dan *non-Aedes* menggunakan *Convolution Neural Networks* dan *Extreme Learning Machine*.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari bab 4 adalah sebagai berikut:

1. Model CNN-ELM yang sudah dilatih dengan menggunakan data latih sebanyak 2325 gambar dapat mengklasifikasikan larva *Aedes* dan *non-Aedes* dengan akurasi 98%, *f1-score Aedes* 99%, *f1-score non-Aedes* 96%, hasil ini didapat dengan menggunakan data validasi sebanyak 582 gambar dengan *true positive* sebanyak 424, *false positive* sebanyak 5, *true negative* sebanyak 146, dan *false negative* sebanyak 7.
2. Model CNN-ELM yang menggunakan model CNN yang telah dilatih menghasilkan akurasi 97% dan 98%, sedangkan model CNN-ELM yang menggunakan model CNN yang tidak dilatih menghasilkan akurasi 83% dan 84%.

5.2. Saran

Saran yang dapat diberikan untuk membantu pengembangan sistem selanjutnya adalah sebagai berikut:

1. Sistem yang dikembangkan hanya dapat mengklasifikasikan larva *Aedes* dan *non-Aedes*, sebaiknya ditambah jenis yang dapat diklasifikasikan.
2. Menambah tipe file yang dapat digunakan untuk mengklasifikasikan larva nyamuk selain gambar.
3. Sebaiknya menggunakan data dengan resolusi yang lebih besar agar dapat diklasifikasi dengan lebih tepat.

DAFTAR PUSTAKA

- Akusok, A., Björk, K. M., Miche, Y., & Lendasse, A. (2015). High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access*, 3, 1011-1025.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2623-2631).
- Angra, S., & Ahuja, S. (2017). Machine learning and its applications: a review. In *2017 International Conference On Big Data Analytics and computational Intelligence (ICBDACI)*, 57-60. IEEE.
- Azman, M. I. A. B. Z., & Sarlan, A. B. (2020). Aedes larvae classification and detection (ALCD) system by using deep learning. In *2020 International conference on computational intelligence (ICCI)* (pp. 179-184). IEEE.
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.
- Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform* (pp. 59-64). Berkeley, CA: Apress.
- Bhutani, K. R., & Farsaie, A. (1991). Fuzzy approach to a neural network. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks* (pp. 1675-1680). IEEE.
- Bock, S., Goppold, J., & Weiß, M. (2018). An improvement of the convergence proof of the ADAM-Optimizer. *arXiv preprint arXiv:1804.10587*.
- Chen, E., Wu, X., Wang, C., & Du, Y. (2019). Application of improved convolutional neural network in image classification. In *2019 International conference on machine learning, big data and business intelligence (MLBDBI)* (pp. 109-113). IEEE.

- De Silva, W. D. M., & Jayalal, S. (2020). Dengue mosquito larvae identification using digital images. In *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)* (pp. 31-36). IEEE.
- Ebert, C., & Louridas, P. (2016). Machine learning. *IEEE Software*, 33(5), 110-115.
- Gangolli, S. H., Fonseca, A. J. L., & Sonkusare, R. (2019). Image enhancement using various histogram equalization techniques. In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-5). IEEE.
- Gülbaş, B., Şengür, A., İncel, E., & Akbulut, Y. (2019). Deep features and extreme learning machines based apparel classification. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-4). IEEE.
- Harish, B. S., Maheshan, M. S., & Nagadarshan, N. (2020). A convolution neural network engine for sclera recognition.
- Kannoja, S. P., & Jaiswal, G. (2018). Ensemble of hybrid CNN-ELM model for image classification. In *2018 5th international conference on signal processing and integrated networks (SPIN)* (pp. 538-541). IEEE.
- Kementerian Kesehatan Republik Indonesia. (2022, September 23). Masuk Peralihan Musim, Kemenkes Minta Dinkes Waspada Lonjakan DBD. Sehatnegeriku:
<https://sehatnegeriku.kemkes.go.id/baca/umum/20220923/3741130/masuk-peralihan-musim-kemenkes-minta-dinkes-waspada-lonjakan-dbd/#:~:text=Berdasarkan%20catatan%20dari%20Direktorat%20Pencegahan,umum%20terjadi%20peningkatan%20kasus%20Dengue.>
- Khellal, A., Ma, H., & Fei, Q. (2018). Convolutional neural network features comparison between back-propagation and extreme learning machine. In *2018 37th Chinese Control Conference (CCC)* (pp. 9629-9634). IEEE.
- Khellal, A., Ma, H., & Fei, Q. (2018). Convolutional neural network based on extreme learning machine for maritime ships recognition in infrared images. *Sensors*, 18(5), 1490.

- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lacerda, P., Barros, B., Albuquerque, C., & Conci, A. (2021). Hyperparameter optimization for COVID-19 pneumonia diagnosis based on chest CT. *Sensors*, 21(6), 2174.
- Nahiduzzaman, M., Goni, M. O. F., Anower, M. S., Islam, M. R., Ahsan, M., Haider, J., & Islam, M. R. (2021). A novel method for multivariant pneumonia classification based on hybrid CNN-PCA based feature extraction using extreme learning machine with CXR images. *IEEE Access*, 9, 147512-147526.
- Nurlaela, N., Niswar, M., Nurtanio, I., Fujaya, Y., Kashihara, S., & Fall, D. (2019). Detection of Megalopa Phase Crab Larvae Using Digital Image Processing. In *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 269-272). IEEE.
- Patro, S. G. K., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.
- Pratama, N. H., Rachmawati, E., & Kosala, G. (2022). CLASSIFICATION OF DOG BREEDS FROM SPORTING GROUPS USING CONVOLUTIONAL NEURAL NETWORK. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(4), 1080-1087.
- Rawat, S. S., Bisht, A., & Nijhawan, R. (2019). A Deep Learning based CNN framework approach for Plankton Classification. In *2019 Fifth International Conference on Image Information Processing (ICIIP)* (pp. 268-273). IEEE.
- Reddy, R. V. K., Raju, K. P., Kumar, L. R., & Kumar, M. J. (2016). Grey level to RGB using YCbCr color space technique. *International Journal of Computer Applications*, 147(7).
- Rodrigues, I. R., da Silva Neto, S. R., Kelner, J., Sadok, D., & Endo, P. T. (2021). Convolutional extreme learning machines: a systematic review. In *Informatics* (Vol. 8, No. 2, p. 33). MDPI.

- Samat, A., Du, P., Liu, S., Li, J., & Cheng, L. (2014). E²LMs: Ensemble Extreme Learning Machines for Hyperspectral Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4), 1060-1069.
- Sanchez-Ortiz, A., Fierro-Radilla, A., Arista-Jalife, A., Cedillo-Hernandez, M., Nakano-Miyatake, M., Robles-Camarillo, D., & Cuatrecasas-Jiménez, V. (2017). Mosquito larva classification method based on convolutional neural networks. In *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (pp. 1-6). IEEE.
- Sesulihatien, W. T., Yuana, D. B. M., & Basuki, A. (2020). Kinematic Feature for Classifying Larvae: Aedes Larvae and Culex Larvae. In *2020 International Electronics Symposium (IES)* (pp. 649-654). IEEE.
- Shekhar, S., Bansode, A., & Salim, A. (2021). A comparative study of hyperparameter optimization tools. In *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)* (pp. 1-6). IEEE.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Singh, S. A., Meitei, T. G., & Majumder, S. (2020). Short PCG classification based on deep learning. In *Deep Learning Techniques for Biomedical and Health Informatics* (pp. 141-164). Academic Press.
- Tato, A., & Nkambou, R. (2018). Improving adam optimizer.
- Tiwari, V., Pandey, C., Dwivedi, A., & Yadav, V. (2020). Image classification using deep neural network. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (pp. 730-733). IEEE.
- Varma, K., Nyman, L., Tountas, K., Sklivanitis, G., Nayak, A. R., & Pados, D. A. (2020). Autonomous Plankton Classification from Reconstructed Holographic Imagery by L1-PCA-assisted Convolutional Neural Networks. In *Global Oceans 2020: Singapore–US Gulf Coast* (pp. 1-6). IEEE.

- Wang, L., You, Z. H., Huang, D. S., & Zhou, F. (2018). Combining high speed ELM learning with a deep convolutional neural network feature encoding for predicting protein-RNA interactions. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(3), 972-980.
- Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, 10(5), 1897.
- World Health Organization (2020). Dengue and Severe Dengue: <https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue>.
- Yoo, Y., & Oh, S. Y. (2016). Fast training of convolutional neural network classifiers through extreme learning machines. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 1702-1708). IEEE.
- Zhang, H., Zhu, Y., Liu, Z., Peng, Y., Peng, W., Tong, L., ... & Cheng, G. (2022). A volatile from the skin microbiota of flavivirus-infected hosts promotes mosquito attractiveness. *Cell*, 185(14), 2510-2522.