

## Evaluasi Kinerja Komputer Mikro Raspberry Pi dengan Pembelajaran Mesin Untuk Pengenalan Wajah

Denis Prayogi

Teknik Informatika, STMIK PPKIA Tarakanita Rahmawati, Tarakan, Indonesia

E-mail: denis@ppkia.ac.id

(\* : corresponding author)

### Abstrak

Implementasi sistem pengenalan wajah pada perangkat tertanam seringkali dibatasi oleh sumber daya komputasi yang terbatas dan biaya hardware yang tinggi. Studi ini bertujuan untuk mengevaluasi dan membandingkan kinerja algoritma Support Vector Machine (SVM) dan Convolutional Neural Network (CNN) pada mikrokomputer Raspberry Pi 4B. Metode penelitian melibatkan pengujian arsitektur CNN dan kernel SVM pada dataset Kaggle yang terdiri dari 2.000 gambar wajah dari lima kelas identitas. Parameter evaluasi yang digunakan meliputi akurasi, presisi, recall, F-Score, penggunaan sumber daya (CPU/RAM), dan kecepatan inferensi. Hasil pengujian menunjukkan bahwa algoritma CNN mencapai akurasi tertinggi sebesar 93%, namun memerlukan waktu inferensi yang lebih lama, yaitu rata-rata 268.52 ms per citra. Di sisi lain, SVM mencapai akurasi 87% dengan waktu inferensi yang jauh lebih cepat, yaitu rata-rata 8.02 ms per citra. Berdasarkan hasil uji, studi ini menyimpulkan bahwa meskipun CNN unggul dalam akurasi, SVM lebih direkomendasikan untuk aplikasi sistem biometrik real-time pada mikrokomputer karena efisiensi waktu komputasi dan penggunaan sumber daya yang lebih rendah.

**Kata kunci:** CNN, Pembelajaran mesin, Pengenalan wajah, Raspberry Pi, SVM.

### Abstract

*Limited computing resources and high hardware costs often limit the implementation of facial recognition systems on embedded devices. This study aims to evaluate and compare the performance of Support Vector Machine (SVM) and Convolutional Neural Network (CNN) algorithms on a Raspberry Pi 4B microcomputer. The research method involves testing the CNN architecture and SVM kernel on a Kaggle dataset consisting of 2,000 facial images from five identity classes. The evaluation parameters used include accuracy, precision, recall, F-Score, resource usage (CPU/RAM), and inference speed. The test results show that the CNN algorithm achieves 93% accuracy but takes longer inference time, averaging 268.52 ms per image. On the other hand, SVM achieves 87% accuracy with much faster inference time, averaging 8.02 ms per image. Based on the test results, this study concludes that although CNN is superior in accuracy, SVM is more recommended for real-time biometric system applications on microcomputers due to its computational time efficiency and lower resource usage.*

**Keywords:** CNN, Face Recognition, Machine Learning, Raspberry Pi, SVM.

## 1. PENDAHULUAN

Dalam beberapa tahun terakhir, keamanan identitas berbasis biometrik telah berkembang pesat dan digunakan secara luas untuk identifikasi dan verifikasi [1]. Biometrik adalah identitas biologis yang unik dari seseorang, seperti sidik jari, suara, geometri tangan, retina mata, dan wajah. Biometrik merupakan solusi yang andal dan cepat, serta paling sesuai untuk mengidentifikasi dan memverifikasi identitas guna mencegah penipuan atau pencurian [2]. Di antara berbagai identitas biologis tersebut, salah satu yang paling populer dan relevan yang diterapkan adalah pengenalan wajah, karena teknologi ini banyak digunakan pada tingkat aplikasi di masyarakat [3] dalam kehidupan sehari-hari, sistem kontrol akses cerdas, dan berbagai bidang. Sistem pengenalan wajah adalah teknologi yang dapat mengenali seseorang berdasarkan gambar digital atau frame video dengan membandingkan fitur wajah gambar yang diberikan dengan gambar dalam database [4]. Fitur kecerdasan buatan berbasis pembelajaran mesin juga diterapkan untuk mengenali seseorang dengan menganalisis pola, tekstur, dan bentuk wajah menggunakan gambar pelatihan per orang [5]. Pengenalan wajah dapat dikategorikan sebagai teknik prediksi dan klasifikasi menggunakan gambar, sehingga sangat cocok jika solusi menggunakan algoritma pembelajaran mesin (ML). ML adalah sistem cerdas yang dapat mengotomatisasi proses dengan mempelajari hubungan pola yang bermakna untuk tugas-tugas spesifik [6]. ML dibangun melalui

proses pelatihan dengan menghasilkan model analitis untuk melakukan tugas-tugas seperti deteksi [7], prediksi, klasifikasi [8], dan pengenalan pola [9].

Beberapa contoh studi sebelumnya, seperti Chandirasekaran dan Shridevi pada tahun 2022 [10] melakukan penelitian terkait ML dengan mengaplikasikan klasifikasi gambar pada Smart Elevator Obstruction. Menggunakan algoritma Support Vector Classifier (SVC), Decision Tree, k-Nearest Neighbor, Random Forest, dan CNN, dengan total data sampel sebanyak 10.000 gambar. Hasil menunjukkan bahwa k-NN dan CNN memiliki akurasi tertinggi meskipun waktu komputasi kedua algoritma ini paling lama. Fadlil dkk [11] membandingkan kinerja pembelajaran mesin untuk mengklasifikasi sampah botol plastik. Algoritma CNN dan SVM digunakan pada penelitian ini, kinerja CNN lebih baik dari segi akurasi dengan mencapai 99% dibandingkan dengan SVM yang hanya 74%.

SVM dan CNN merupakan algoritma pembelajaran mesin yang paling populer di antara beberapa algoritma lainnya [11] dan unggul dibandingkan algoritma lain yang digunakan untuk klasifikasi menggunakan gambar. SVM adalah algoritma berbasis kernel untuk menyelesaikan kasus klasifikasi dan regresi linier, yang diperkenalkan oleh Vapnik dan Cortes. Algoritma ini efektif dalam menyelesaikan masalah klasifikasi dalam pembelajaran mesin terawasi karena kemampuannya yang baik dalam generalisasi, terutama pada data input yang kecil, dan akurasi yang dihasilkan lebih baik daripada algoritma lain. Namun, SVM juga memiliki kelemahan, seperti tidak cocok untuk data input yang besar karena kompleksitas algoritma ini, masalah klasifikasi multi-kelas, dan data input yang tidak seimbang yang menyebabkan akurasi menurun. CNN adalah salah satu algoritma pembelajaran mendalam berbasis jaringan saraf konvensional berlapis-lapis yang digunakan untuk menangani masalah klasifikasi dan pengenalan gambar yang baik karena menghasilkan akurasi yang baik. Dalam beberapa tahun terakhir algoritma ini mengalami kemajuan yang signifikan dalam pengolahan citra dibanding metodologi tradisional pembelajaran mesin [12]. CNN dapat digunakan untuk pengenalan wajah, kendaraan otonom, layanan checkout cerdas, pengobatan medis berbasis kecerdasan buatan [13], anti-spoofing, dan Gambar Hiper-Spektral (HSI) [14]. Algoritma CNN cocok untuk data input yang besar menggunakan Unit Pemrosesan Grafis (GPU) tetapi memerlukan waktu pelatihan yang lama [15]. SVM dan CNN menggunakan teknik klasifikasi yang berbeda, dan hasil akurasi selalu dinamis berdasarkan studi sebelumnya [11]. Membangun platform dengan kedua algoritma ini memerlukan perangkat berkemampuan tinggi dan biaya yang besar, terutama dalam pembelajaran mendalam seperti CNN jika dijalankan dengan perangkat dengan sumber daya terbatas [16].

Implementasi sistem pengenalan wajah pada perangkat *edge computing* menjadi solusi karena mampu menekan biaya operasional dan meningkatkan responsivitas sistem secara real-time [16]. Oleh karena itu, penelitian ini menganalisis perbandingan kinerja algoritma SVM dan CNN pada platform Raspberry Pi 4B. Pemilihan perangkat ini merujuk pada penelitian [17] yang menyatakan bahwa Raspberry Pi 4B memiliki keunggulan performa yang jauh lebih tinggi dibandingkan model-model Raspberry Pi lainnya. Penggunaan mikrokomputer ini bertujuan menghasilkan sistem keamanan biometrik yang lebih efisien dari sisi biaya dan konsumsi energi. Untuk menguji efektivitas arsitektur tersebut, penulis menggunakan dataset wajah beresolusi 200x200 piksel dari Kaggle.

## **2. METODE PENELITIAN**

Mikrokomputer adalah komputer portabel berbiaya rendah dengan harga terjangkau dibandingkan komputer biasa. Salah satu jenis mikrokomputer yang sering digunakan adalah Raspberry Pi karena bisa digunakan untuk penelitian [18]. Raspberry Pi memiliki beberapa tipe, termasuk Zero, 3B atau 3B+, dan 4B. Diantara berbagai tipe tersebut, 4B merupakan salah satu yang unggul dalam spesifikasi dan kemampuan [17]. Raspberry Pi 4B yang digunakan dalam penelitian ini dilengkapi dengan prosesor ARM v8 Quad-core CortexA72 berkecepatan 1.5 GHz, memori 2 GB [19], dan GPU terintegrasi 32-bit Broadcom Video Core VI. Sistem operasi yang digunakan untuk membangun ML pada Raspberry Pi adalah versi 11 Bullseye 64-bit dengan Python 3.9.2, editor Jupyter Notebook versi 6.4.10, OpenCV 4.5.5, Skicit-learn 1.0.2, Keras 2.8.0,

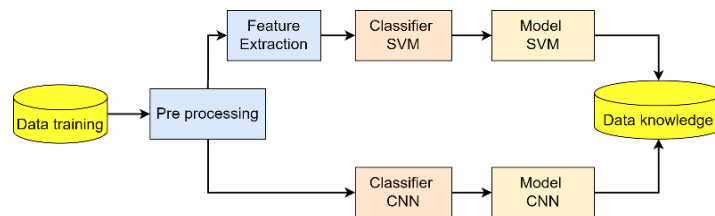
dan TensorFlow 2.8.0.

Contoh gambar yang digunakan dalam makalah ini adalah wajah lima (5) orang dengan berbagai ekspresi, yang diambil secara acak dari dataset Kaggle dengan total 2000 contoh gambar. Gambar 1 adalah contoh gambar wajah 5 orang yang digunakan.



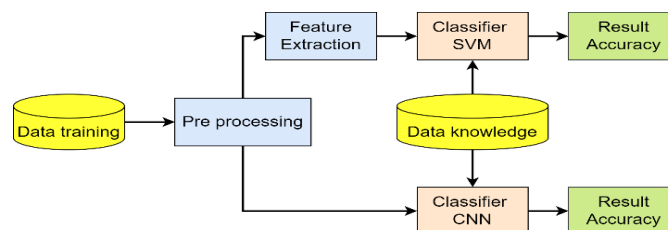
Gambar 1. Sampel Wajah

Setiap gambar diberi label berdasarkan nama, kemudian sampel dibagi dengan komposisi 80% data pelatihan dan 20% data uji sebesar dan perbandingan 90%:10%. Data pelatihan digunakan untuk menghasilkan model dari kedua algoritma, sementara data uji digunakan untuk menguji model guna menentukan akurasi. Alur kerja kedua algoritma terdapat dalam proses pelatihan.



Gambar 2. Alur kerja pelatihan data algoritma CNN dan SVM

Data gambar yang telah diberi label masuk ke tahap pra-pemrosesan. Pada tahap ini, data gambar diperluas untuk meningkatkan jumlah sampel agar model yang dihasilkan memiliki akurasi yang baik, kemudian dimensi gambar dikurangi menjadi 64x64 piksel. Pengurangan ini dilakukan agar proses pelatihan tidak memakan waktu terlalu lama karena keterbatasan spesifikasi komputasi Raspberry Pi. Gambar yang telah dikurangi diubah menjadi gambar grayscale untuk menyederhanakan nilai yang terkandung dalam setiap piksel. Konsep awal yang terdiri dari tiga lapisan (Red, Green, dan Blue/RGB) menjadi satu lapisan setelah diubah menjadi grayscale. Gambar 2 menunjukkan alur proses yang berbeda dalam pelatihan data. SVM dalam studi ini menggunakan ekstraksi fitur PCA berbasis eigenface, sementara ekstraksi fitur CNN terjadi dalam proses tersebut. Ekstraksi fitur bertujuan untuk menentukan karakteristik setiap gambar. Pelatihan dua algoritma akan menghasilkan model yang mengandung basis data pengetahuan yang akan dievaluasi menggunakan data pengujian. Gambar 3 adalah alur kerja proses pengujian basis data pengetahuan.



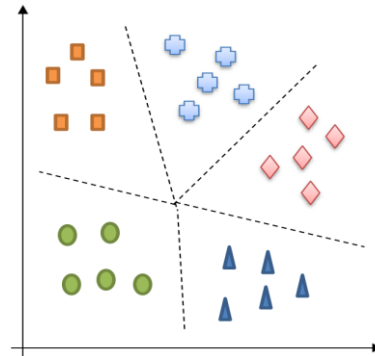
Gambar 3. Alur kerja algoritma CNN dan SVM

Pengetahuan data dari proses pelatihan SVM dan CNN akan diuji menggunakan data uji untuk mengevaluasi model, sehingga tingkat akurasi dapat ditentukan. Selain melihat akurasi, pengukuran kinerja model dapat diketahui berdasarkan penjelasan pada bagian evaluasi dalam makalah ini.

## 2.1 Support Vector Machine

SVM merupakan metode pembelajaran mesin terawasi dan sering digunakan untuk kasus klasifikasi maupun regresi. Untuk kasus klasifikasi, digunakan Support Vector Classifier (SVC),

sedangkan untuk kasus regresi digunakan Support Vector Regression (SVR). SVM dapat menggunakan vektor pendukung atau subset dari titik pelatihan. Metode ini lebih efisien dalam penggunaan memori, efektif jika data pelatihan lebih sedikit daripada data sampel, dan efektif dalam ruang dimensi tinggi. SVM menggunakan garis hiperplane untuk memisahkan batas-batas setiap kelas pada ruang berdimensi tinggi saat melatih data. Gambar 4 menunjukkan ilustrasi klasifikasi dengan mengoptimalkan hiperplane pada *multiclass* sebanyak lima kelas data yang digunakan pada penelitian ini [20].

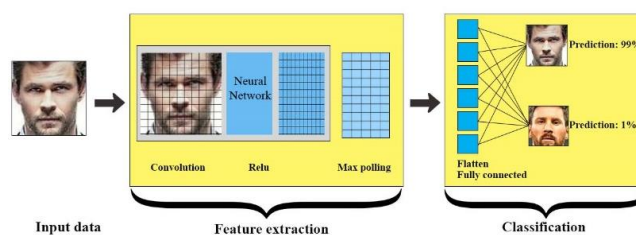


Gambar 4. Hiperplane *multiclass* SVM

Mengoptimalkan margin antara kelas, SVM menggunakan kernel, yaitu fungsi kemiripan untuk menemukan kemiripan dalam data. Dalam penelitian ini, pustaka yang digunakan adalah scikit-learn, dengan kernel Polynomial dan Radial Basis Function (RBF), dengan mengoptimalkan parameter biaya (C) dan gamma. C adalah parameter bias yang digunakan untuk meminimalkan kesalahan klasifikasi, sementara gamma mengontrol jarak antara titik vektor selama pelatihan data.

## 2.2 Convolutional Neural Network

CNN merupakan algoritma pembelajaran mesin dalam yang diawasi. CNN menggunakan jaringan saraf tiruan feed-forward berlapis-lapis, dapat mengekstrak fitur dari data dengan struktur konvolusi. CNN banyak digunakan dalam pemrosesan gambar dan video, seperti deteksi dan klasifikasi objek. CNN memiliki berbagai arsitektur jaringan saraf tiruan, seperti jaringan saraf tiruan rekurens (RNN) [21], EfficientNet-B7 [22], CNN berbasis wilayah (R-CNN) [23], MOPSO-CNN (MOCNN), VGG16, VGG 19 [24], ResNet50, MobileNetV2, dan DenseNet169 [25].



Gambar 5. Alur kerja proses CNN

CNN bekerja menggunakan teknik konvolusi dengan dua tahap, yaitu ekstraksi fitur dan klasifikasi. Ekstraksi fitur terdiri dari data input yang mewakili data matriks gambar dan kemudian diabstrakkan menjadi nilai lapisan fitur peta konvolusi dua dimensi. Output lapisan konvolusi adalah kernel yang terdiri dari array nilai. Nilai array tersebut kemudian dikurangi ukurannya menggunakan Max pooling. Tujuannya adalah untuk menghasilkan lapisan baru yang berisi nilai piksel paling signifikan dari lapisan sebelumnya. Teknik ini dapat diulang untuk meningkatkan jumlah neuron dan kombinasi dalam lapisan konvolusi. Nilai-nilai array dua dimensi kemudian diratakan menjadi array satu dimensi berupa nilai vektor, dimulai dari sisi kanan atas ke sisi kiri bawah dalam array dua dimensi. Array satu dimensi yang dihasilkan

dimasukkan ke dalam lapisan terhubung sepenuhnya. Tahap akhir proses CNN adalah klasifikasi untuk mengukur prediksi dari gambar input berdasarkan nilai array yang dihasilkan dari ekstraksi fitur.

### 2.3 Evaluasi

Model yang dihasilkan dari proses pelatihan dan pengujian dua algoritma diukur menggunakan perhitungan evaluasi. Beberapa indikator evaluasi digunakan untuk menguji efektivitas pembelajaran mesin, yaitu akurasi, *recall*, *F-Score*, dan presisi [26]. Akurasi untuk menentukan persentase data prediksi yang benar dari total data. Presisi digunakan untuk menentukan persentase data aktual yang benar dari total prediksi. Recall (sensitivitas) digunakan untuk menentukan persentase data prediksi yang benar dibandingkan dengan total data aktual yang diprediksi. Pada saat yang sama, *F-Score* digunakan untuk membandingkan rata-rata presisi dan recall, yang diweighted. Hasil evaluasi akan ditampilkan dalam bentuk confusion matrix dan tabel yang digunakan untuk memvisualisasikan kinerja algoritma pembelajaran terawasi [27]. Evaluasi dihitung berdasarkan persamaan:

$$accuracy = \frac{TP}{Total\ Data} \quad (1)$$

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

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

$$F - Score = \frac{2 \times (precision \times recall)}{(precision + recall)} \quad (4)$$

Dimana:

TP: True Positive, Kondisi dimana model memprediksi positif, kenyataannya positif

FP: False Positive, Kondisi dimana model memprediksi positif, kenyataannya negatif

FN: False Negative, Kondisi dimana model memprediksi negatif, kenyataannya positif

Pengukuran juga dilakukan pada Raspberry Pi untuk menentukan kemampuannya menjalankan machine learning. Uji benchmark menggunakan perangkat lunak Htop yang dapat menampilkan kinerja Raspberry Pi, seperti CPU, RAM, dan sumber daya lainnya di papan tersebut.

## 3. HASIL DAN PEMBAHASAN

Eksperimen pertama menggunakan dataset berupa 1000 data gambar yang dibagi menjadi lima kelas dari data Kaggle [28]. Artinya, setiap kelas data memiliki 200 sampel gambar wajah individu. Kemudian, dari empat percobaan eksperimental, sampel data ditingkatkan dari 200 kelas masing-masing menjadi 400 data atau total 2000 gambar untuk empat percobaan berikutnya. Sampel data direproduksi menggunakan teknik augmentasi, yaitu memutar objek gambar secara horizontal dengan sudut 25°. Augmentasi juga mengubah kecerahan objek gambar menjadi lebih gelap dan lebih terang, dengan rentang nilai 0,5–1,5. Penyesuaian tingkat kecerahan ini sangat penting dalam melatih model karena sebagian besar objek gambar tidak berada dalam pencahayaan yang sempurna. Data gambar direproduksi untuk menentukan peningkatan akurasi algoritma pembelajaran mesin, waktu pelatihan, dan penggunaan sumber daya hardware.

### 3.1 SVM

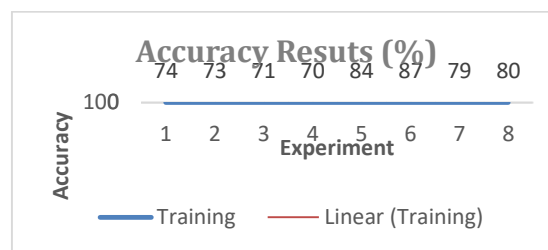
Arsitektur SVM yang dibangun berdasarkan pembelajaran mesin pada Raspberry Pi menggunakan kernel Fungsi Basis Radial (RBF) dan polinomial untuk kasus klasifikasi multi-kelas. Dengan menggunakan pencarian grid SVC, hiperparameter C dan gamma dapat dimaksimalkan dengan mencari nilai kandidat secara otomatis untuk menghasilkan akurasi terbaik. Pengaturan hiperparameter adalah  $C = \{1000, 5000, 10000, 500000, 100000\}$ , sedangkan  $\gamma = \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1\}$ . Eksperimen pertama hingga keempat menggunakan 1000 gambar dengan komposisi 80% data pelatihan dan 20% data uji sebesar dan perbandingan 90%:10%. Kemudian pada eksperimen kelima hingga kedelapan, data gambar

ditingkatkan menjadi 2000. Tabel 1 menampilkan hasil kinerja klasifikasi menggunakan SVM.

Tabel 1. Hasil Performa Algoritma SVM

No	Data	Komposisi (%)	Kernel	C	Gamma	Recall (%)	Precision (%)	F-Score (%)	Accuracy (%)	Time (s)
Exp1	1000	80:20	RBF	1000	0.100	74	74	74	74	75
Exp2		90:20		10000	0.001	74	74	73	73	96
Exp3		80:20	Polynomial	5000	0.010	71	74	71	71	11
Exp4		90:20		50000	0.005	70	75	71	70	16
Exp5	2000	80:20	RBF	1000	0.100	84	84	84	84	411
Exp6		<b>90:10</b>		<b>1000</b>	<b>0.100</b>	<b>87</b>	<b>87</b>	<b>87</b>	<b>87</b>	<b>527</b>
Exp7		80:20	Polynomial	5000	0.010	79	79	79	79	48
Exp8		90:10		5000	0.01	80	82	80	80	61

Dari Tabel 1, nilai C dan gamma bervariasi. Nilai-nilai ini merupakan kandidat terbaik selama eksperimen dengan menghasilkan recall terbaik berdasarkan Persamaan (3), *precision* terbaik berdasarkan Persamaan (2), dan *F-Score* terbaik berdasarkan Persamaan (4). Dari delapan percobaan, skor terendah adalah 70%-75% pada Exp1-Exp4 dengan total data 1000 gambar. Namun, waktu pelatihan rata-rata lebih singkat dibandingkan dengan Exp5-Exp8 yang menggunakan 2000 data. Berdasarkan persamaan (1), Gambar 6 adalah grafik dari akurasi terbaik yang didapatkan saat pelatihan menggunakan kernel RBF, hiperparameter C=1000, dan Gamma=0.100.



Gambar 6. Hasil Akurasi SVM

Dari grafik pada Gambar 6, garis tren tidak menunjukkan peningkatan yang signifikan, dan nilainya masih di bawah 90%. Prestasi terbaik SVM terdapat pada Exp6, dengan akurasi tertinggi sebesar 87%. Dalam hal ini, SVM memerlukan lebih banyak data selama proses pelatihan untuk mencapai akurasi yang lebih baik. Exp6 adalah eksperimen yang menghasilkan nilai terbaik, dengan matriks pada Tabel 2.

Tabel 2. Confusion Matrix SVM

	Class	Prediction Label				
		Dwayne Johnson	Chris Hemsworth	Rebecca Ferguson	Lindsey Morgan	Lionel Messi
True Label	Dwayne Johnson	38	2	0	1	0
	Chris Hemsworth	1	30	4	1	0
	Rebecca Ferguson	2	1	35	2	0
	Lindsey Morgan	2	0	3	31	1
	Lionel Messi	2	1	1	2	40

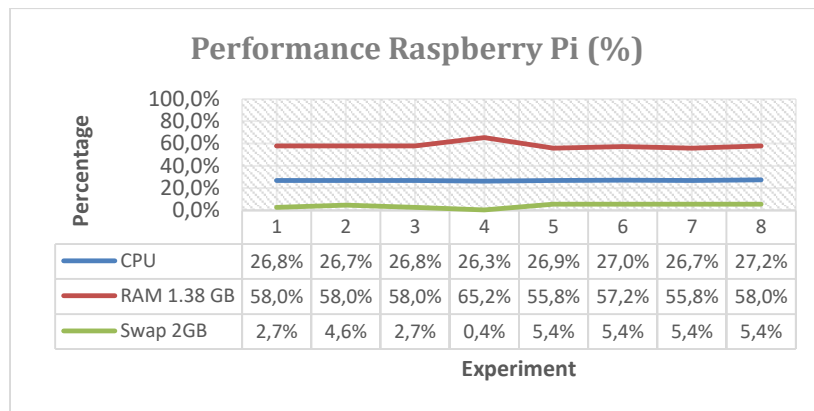
Matriks Exp6 menampilkan beberapa data benar dan salah dalam klasifikasi pengenalan wajah pada data uji. Data aktual menunjukkan kelas sebenarnya, sementara data prediksi menunjukkan hasil dari algoritma SVM. Dari Tabel 2, TP adalah jumlah total data yang diprediksi dengan benar (blok ungu), sebanyak 174. Jumlah total data uji dalam Exp6 adalah 200 data. Kemudian berdasarkan persamaan (1), maka hasil akurasi adalah 87%.

Dalam kasus ini, FP diilustrasikan pada kelas "Dwayne Johnson". Terdapat kesalahan

prediksi data, di mana data aktual bukanlah “Dwayne Johnson” tetapi diprediksi sebagai “Dwayne Johnson” (ditunjukkan oleh panah biru). Seperti yang dapat dilihat, kesalahan prediksi “Chris Hemsworth” satu data, “Rebecca Ferguson”, “Lindsey Morgan”, “Lionel Messi” masing-masing dua data. Jika total dari lima kelas adalah, maka  $FP = 26$ . Kemudian berdasarkan persamaan (2), maka hasil dari presisi adalah 87%.

FN adalah semua data yang diprediksi sebagai salah. Misalnya, dalam kelas “Dwayne Johnson”, terdapat kesalahan prediksi data, yaitu “Dwayne Johnson” yang diprediksi sebagai “Chris Hemsworth” sebanyak dua data dan “Lindsey Morgan” sebanyak satu data (ditunjukkan oleh panah hijau). Jika total dari lima kelas adalah, maka  $FN = 26$ . Kemudian berdasarkan persamaan (3), maka hasil dari *recall* adalah 87%. Jadi, *F-Score* dapat dihitung menggunakan persamaan (4), yaitu sebesar 87%.

Dari perhitungan akurasi, presisi, recall, dan *F-Score*, hasilnya terdapat pada Tabel 1 untuk Exp6. Kesalahan tertinggi terdapat pada kelas “Chris Hemsworth”, yang diprediksi sebagai “Rebecca Ferguson” dengan empat gambar. Pada saat yang sama, data benar tertinggi terdapat pada kelas “Lionel Messi” dengan 40 data gambar. Kesalahan akurasi keseluruhan prediksi adalah 13%. Penggunaan sumber daya Raspberry selama pelatihan menunjukkan angka yang cukup baik. Pengukuran yang dilakukan menggunakan perangkat lunak Htop pada rata-rata CPU, RAM, dan swap memori dapat dilihat pada Gambar 7.



Gambar 7. Penggunaan Sumber Daya Raspberry Pi Selama Pelatihan SVM

Penggunaan CPU selama pelatihan data SVM menunjukkan rata-rata rendah sebesar 26,8%, rata-rata RAM sebesar 58,2%, dan rata-rata pertukaran memori sebesar 4,0% selama delapan kali percobaan. Pelatihan SVM mengonsumsi sumber daya umum, terutama CPU, yang bahkan tidak mencapai 30%, sehingga pembelajaran mesin ini sangat cocok untuk perangkat Raspberry Pi. Sayangnya, pengukuran GPU tidak dapat dilakukan karena keterbatasan driver Broadcom, yang merupakan sumber tertutup dari perangkat ini.

### 3.2 CNN

Arsitektur CNN yang digunakan untuk membangun model pada penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Arsitektur CNN

Parameter	CNN-I	CNN-II
Basis Arsitektur	Custom	MobileNetV2
Input Resolution	200 x 200 (RGB)	200 x 200 (RGB)
Hidden Layers	3 Lapis (64, 128, 256)	3 Lapis (128, 64, 32)
Filter Kernel	3 x 3	3 x 3
Fungsi Aktivasi	ReLU	ReLU
Lapisan Klasifikasi	Softmax	Softmax
Optimizer	Adam	Adam
Epoch	30	30

Arsitektur CNN untuk membangun model adalah CNN-I dan CNN-II, dengan 30 iterasi

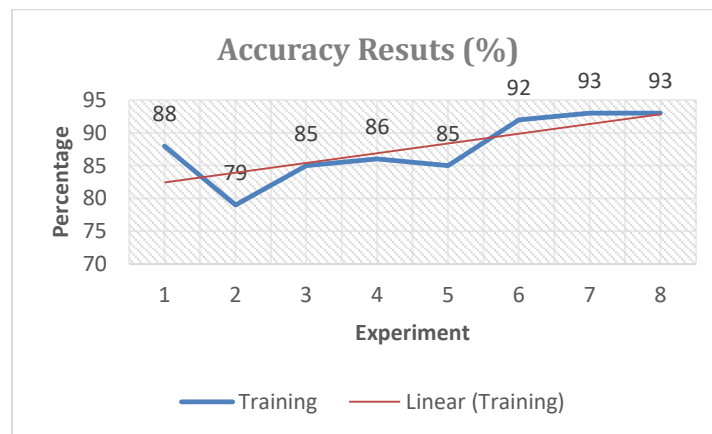


epoch. Perbedaan antara keduanya adalah CNN-I, yang memiliki tiga lapisan tersembunyi berukuran 64x64, 128x128, dan 256x256. Pada saat yang sama, berdasarkan arsitektur MobileNetV2, CNN-II terdiri dari tiga lapisan tersembunyi dengan ukuran 128x128, 64x64, dan 32x32. CNN-I dan CNN-II menggunakan filter kernel 3x3, serta Rectified Linear Unit (ReLU) nonlinier sebagai fungsi aktivasi [7], menggunakan softmax sebagai lapisan klasifikasi [29], dan menggunakan optimizer Adam. Eksperimen pertama hingga keempat menggunakan 1000 gambar dengan komposisi 80% data pelatihan dan 20% data uji sebesar dan perbandingan 90%:10%. Kemudian eksperimen kelima hingga kedelapan menggunakan komposisi data yang sama, tetapi data gambar ditingkatkan menjadi 2000. Tabel 4 adalah hasil aktivitas menggunakan kedua arsitektur CNN.

Tabel 4. Hasil Performa Algoritma CNN

No	Data	Composition (%)	Architecture	Recall (%)	Precision (%)	F-Score (%)	Accuracy (%)	Time (s)
Exp1	1000	80:20	CNN-I	88	88	88	88	3575
Exp2		90:20		79	81	79	79	4217
Exp3		80:20	CNN-II	85	85	85	85	4827
Exp4		90:20		86	87	86	86	5877
Exp5	2000	80:20	CNN-I	85	85	85	85	6886
Exp6		90:10		92	92	92	92	8692
Exp7		80:20	CNN-II	93	93	93	93	9593
Exp8		90:10		93	93	93	93	11869

Berdasarkan Tabel 4, hasil evaluasi menunjukkan kinerja yang berbeda-beda tergantung pada arsitektur CNN dan komposisi data pelatihan dan pengujian. Jumlah data juga secara signifikan mempengaruhi evaluasi *recall*, *precision*, *F-Score*, dan lama proses pelatihan. Eksperimen 1 hingga Eksperimen 4 menunjukkan hasil rata-rata di bawah 90%. Sementara itu, Exp5 hingga Exp8, dengan jumlah gambar dua kali lipat, menunjukkan peningkatan kinerja dengan hasil rata-rata di atas 90%, tetapi durasi waktu pelatihan juga hampir dua kali lipat. Sayangnya, terjadi *overfitting* pada *Train8* pada epoch ke-24 karena arsitektur yang digunakan tidak dapat mempelajari gambar dengan komposisi yang sama, seperti yang ditunjukkan pada Tabel 4. Berdasarkan persamaan (1), hasil akurasi dapat dilihat pada Gambar 8.



Gambar 8. Hasil Akurasi CNN

Hasil akurasi menunjukkan tren peningkatan, dengan akurasi terendah sebesar 79% menggunakan CNN-I pada Exp2 dan akurasi tertinggi sebesar 93% menggunakan arsitektur CNN-II pada Exp7. Tabel 5, yang merupakan matriks kebingungan, menampilkan hasil pengenalan wajah berdasarkan hasil akurasi terbaik pada Exp7.

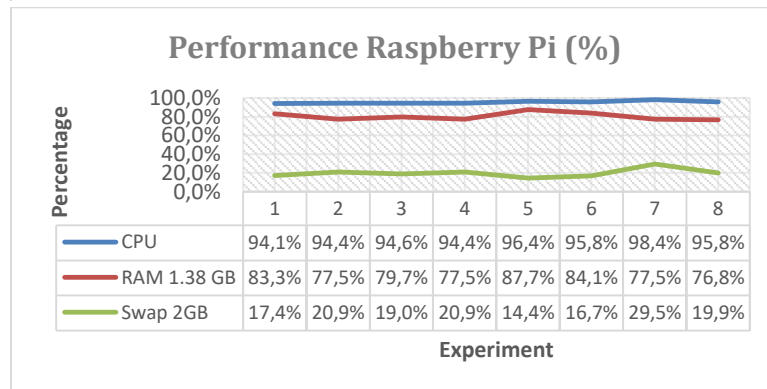


Tabel 5. Confusion Matrix CNN

True Label	Class	Prediction Label				
		Chris Hemsworth	Dwayne Johnson	Lindsey Morgan	Lionel Messi	Rebecca Ferguson
	Chris Hemsworth	63	2	0	0	2
	Dwayne Johnson	3	75	1	1	0
	Lindsey Morgan	1	2	70	0	2
	Lionel Messi	5	1	0	80	0
	Rebecca Ferguson	1	1	7	1	82

Confusion matrix menunjukkan kesalahan klasifikasi dalam pengenalan wajah berdasarkan data uji. Data aktual menunjukkan kelas sebenarnya, sementara data prediksi menunjukkan hasil prediksi dari algoritma CNN. Dari Tabel 5, TP (blok ungu) adalah 370 dari 400 data uji pada Exp7. Total FP (ditunjukkan oleh panah biru) dari 5 kelas adalah 30 dan FN (ditunjukkan oleh panah hijau) adalah 30. Kemudian berdasarkan persamaan (1-4), hasil akurasi adalah 93%, presisi 93%, *recall* 93%, dan *F-Score* 93%:

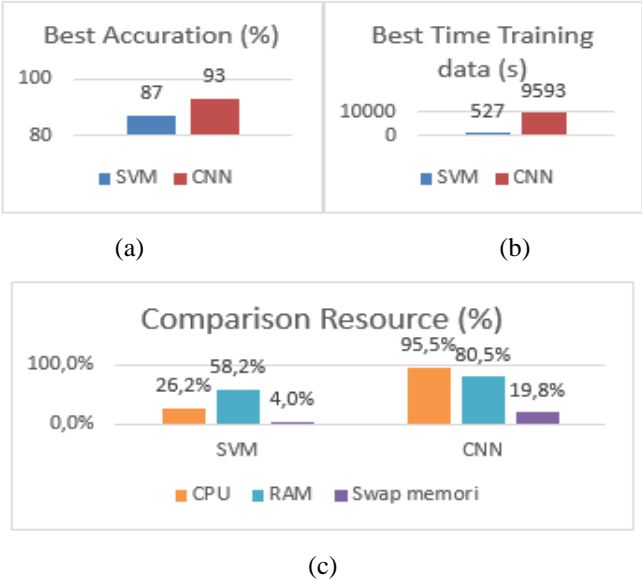
Dari perhitungan akurasi, presisi, *recall*, dan *F-Score*, hasil terbaik terdapat pada Eksperimen 7 yang tercantum dalam Tabel 4. Kesalahan tertinggi terdapat pada kelas “Rebecca Ferguson”, yang diprediksi sebagai “Lindsey Morgan”. Namun, kesalahan prediksi secara keseluruhan hanya sebesar 7%. Meskipun akurasi pelatihan menunjukkan angka yang tinggi, penggunaan sumber daya Raspberry juga menunjukkan angka yang tinggi. Pengukuran dilakukan menggunakan perangkat lunak Htop pada rata-rata CPU, RAM, dan swap memori, seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Penggunaan Sumber Daya Raspberry Pi Selama Pelatihan CNN

Penggunaan CPU rata-rata 95,5%, penggunaan RAM 80,5%, dan Swap memori rata-rata 19,8% selama delapan pelatihan. Penggunaan CPU dan RAM yang tinggi disebabkan oleh proses komputasi yang terjadi pada CNN. Peningkatan penggunaan sumber daya utama seperti CPU dan RAM menyebabkan Raspberry Pi berjalan lambat jika diberi tugas lain.




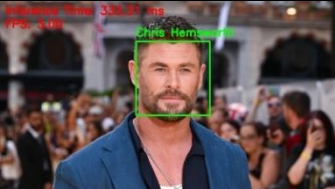
Dalam studi ini, kinerja algoritma CNN lebih baik daripada SVM jika diimplementasikan pada Raspberry Pi. Hasil penelitian menunjukkan bahwa CNN mencapai akurasi terbaik sebesar 93%, sementara SVM hanya 87%. Hasil ini konsisten dengan penelitian [11] yang menyatakan bahwa kinerja CNN lebih baik daripada SVM dalam klasifikasi objek botol sampah. Namun, selain beberapa hasil sebelumnya, hasil penelitian ini tidak serta merta melihat hasil kinerja kedua algoritma. Juga perlu diperhatikan penggunaan sumber daya perangkat, terutama pada mikrokomputer seperti Raspberry Pi, yang memiliki keterbatasan dalam komputasi. Gambar 10 menunjukkan perbandingan keseluruhan kedua algoritma berdasarkan hasil terbaik dalam penelitian ini.



Gambar 10. Perbandingan rata-rata kinerja SVM dan CNN pada Raspberry Pi

Gambar 10(a) membandingkan akurasi kedua algoritma, di mana CNN tampak lebih unggul daripada SVM dalam mengklasifikasikan data. Meskipun ada beberapa kelemahan SVM, seperti data yang tidak seimbang dan pilihan parameter yang terbatas, telah berusaha diperbaiki, hasilnya tidak dapat melebihi CNN pada komposisi data yang sama selama pelatihan. Namun, CNN memiliki kelemahan. Yaitu, waktu komputasi lebih lama daripada SVM, seperti yang ditunjukkan pada Gambar 10(b). Perbedaan waktu pelatihan akibat arsitektur CNN yang menggunakan sistem konvolusi membuat perhitungan berjalan lambat, terutama jika dataset besar. Mengenai penggunaan sumber daya pada Gambar 10(c), SVM sangat direkomendasikan jika diimplementasikan pada Raspberry Pi dibandingkan CNN. SVM lebih efisien dalam penggunaan CPU, dengan rata-rata hanya 26,8% dibandingkan CNN yang hampir sepenuhnya membebani CPU. Raspberry Pi 4B dengan RAM 2GB cukup untuk menjalankan SVM dan CNN, tetapi sekali lagi SVM unggul dengan rata-rata penggunaan 58,2%. Untuk memori swap, pengaturan default pada Raspberry Pi adalah 100 MB, tetapi karena perhitungan CNN memerlukan ruang swap yang lebih besar, memori swap ditingkatkan menjadi 2 GB. Penggunaan memori swap rata-rata CNN adalah 19,8% atau hampir 400 MB per pelatihan, tetapi SVM hanya memerlukan rata-rata 82 MB.

Tabel 6. Skenario Pengujian

SVM Inference time (ms) & FPS	CNN Inference time (ms) & FPS	Keterangan
 9.82 & 101.84	 416.76 & 2.40	SVM memprediksi dengan benar wajah dari Dwayne Johnson
 8.0 & 125.05	 333.31 & 3.00	SVM memprediksi dengan benar wajah dari Chris hemsworth

SVM <i>Inference time (ms) &amp; FPS</i>	CNN <i>Inference time (ms) &amp; FPS</i>	Keterangan
 3.83 & 261.16	 239.39 & 4.18	SVM dan CNN memprediksi dengan benar wajah dari Lionel Messi. Namun perbedaannya terdapat pada inferensi waktu mengenali
 8.01 & 124.84	 282.67 & 3.54	SVM dan CNN memprediksi dengan benar wajah dari Lindsey Morgan. Namun perbedaannya terdapat pada inferensi waktu mengenali
 10.41 & 96.07	 248.23 & 4.03	SVM dan CNN mampu memprediksi dengan benar wajah dari Rebecca Ferguson
 8.06 & 124.09	 152.78 & 6.55	SVM dan CNN memprediksi dengan benar, yaitu dua wajah yang benar, dan satu wajah yang memang tidak ada di kelas data ( <i>unknown</i> ).
 8.01 & 124.92	 206.56 & 4.84	SVM memprediksi keliru, dimana kedua wajah dikenali sebagai Lionel Messi. Sedangkan juga dengan CNN.
Rata-rata ms = <b>8.02</b> FPS = <b>136.85</b>	Rata-rata ms = <b>268.52</b> FPS = <b>4.08</b>	

Selain melihat akurasi dalam proses pelatihan, penelitian ini juga mengevaluasi *inference time* dan *frame per seconds* untuk menguji sistem dalam skenario secara *real-time*. Berdasarkan Tabel 6, meskipun CNN memiliki akurasi pelatihan yang paling tinggi, namun algoritma ini membutuhkan waktu inferensi yang tinggi sebesar rata-rata 268.52 ms. Sebaliknya, SVM menunjukkan keunggulan pada kecepatan proses dengan rata-rata 8.02 ms, sehingga lebih ideal untuk digunakan pada aplikasi navigasi cepat seperti di robotika atau drone.

#### 4. KESIMPULAN DAN SARAN

Studi ini berhasil mengevaluasi algoritma SVM dan CNN pada mikrokomputer Raspberry Pi 4B untuk sistem pengenalan wajah. Berdasarkan hasil uji coba, algoritma CNN dengan arsitektur CNN-II mencapai akurasi tertinggi sebesar 93%, melampaui SVM (kernel RBF) yang mencapai 87%. Namun, efisiensi kinerja menunjukkan bahwa SVM jauh lebih unggul dalam penggunaan sumber daya dan kecepatan pemrosesan (waktu inferensi) dibandingkan CNN, yang memiliki kompleksitas komputasi tinggi. Kesimpulannya, untuk aplikasi yang mengutamakan

akurasi tinggi pada dataset besar, arsitektur CNN-II adalah pilihan terbaik. Sebaliknya, untuk sistem keamanan biometrik *real-time* yang diimplementasikan pada Raspberry Pi dengan daya dan memori terbatas, SVM lebih direkomendasikan karena keseimbangannya antara akurasi yang memadai dan efisiensi sumber daya optimal.

#### DAFTAR PUSTAKA

- [1] M. S. Kabisha, K. A. Rahim, M. Khaliluzzaman, and S. I. Khan, "Face and Hand Gesture Recognition Based Person Identification System using Convolutional Neural Network," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 105–115, 2022, doi: 10.18201/ijisae.2022.273.
- [2] W. Boukhari, M. Benyettou, and B. Abderrahim, "A human vision based system for biometric images recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 3, pp. 1508–1517, 2022, doi: 10.11591/ijeecs.v25.i3.pp1508-1517.
- [3] Y. Aufar and I. S. Sitanggang, "Face recognition based on Siamese convolutional neural network using Kivy framework," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 2, pp. 764–772, 2022, doi: 10.11591/ijeecs.v26.i2.pp764-772.
- [4] Q. Sun and A. Redei, "Knock Knock, Who's There: Facial Recognition using CNN-based Classifiers," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 9–16, 2022, doi: 10.14569/IJACSA.2022.0130102.
- [5] J. Almotiri, "Face Recognition using Principal Component Analysis and Clustered Self-Organizing Map," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 509–520, 2022, doi: 10.14569/IJACSA.2022.0130361.
- [6] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.
- [7] A. Yadav, E. Kumar, and P. K. Yadav, "Object detection and used car price predicting analysis system (UCPAS) using machine learning technique," *Linguist. Cult. Rev.*, vol. 5, no. S2, pp. 1131–1147, 2021, doi: 10.21744/lingcure.v5ns2.1660.
- [8] M. Muslih and E. H. Rachmawanto, "Convolutional Neural Network (Cnn) Untuk Klasifikasi Citra Penyakit Diabetes Retinopathy," *SKANIKA*, vol. 5, no. 2, pp. 167–176, 2022, doi: 10.36080/skanika.v5i2.2945.
- [9] C. Durán *et al.*, "Nonlinear machine learning pattern recognition and bacteria-metabolite multilayer network analysis of perturbed gastric microbiome," *Nat. Commun.*, vol. 12, no. 1, 2021, doi: 10.1038/s41467-021-22135-x.
- [10] P. Chandirasekaran and S. Shridevi, "Smart Elevator Obstruction Detection System using Image Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 134–141, 2022, doi: 10.14569/IJACSA.2022.0130416.
- [11] A. Fadlil, R. Umar, Sunardi, and A. S. Nugroho, "Comparison of Machine Learning Approach for Waste Bottle Classification," *Emerg. Sci. J.*, vol. 6, no. 5, pp. 1075–1085, 2022, doi: 10.28991/ESJ-2022-06-05-011.
- [12] M. Trigka and E. Dritsas, "A Comprehensive Survey of Deep Learning Approaches in Image Processing," *Sensors*, vol. 25, no. 2, 2025, doi: 10.3390/s25020531.
- [13] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2021, doi: <https://doi.org/10.1109/TNNLS.2021.3084827>.
- [14] M. Kanthi, T. H. Sarma, and C. S. Bindu, "Multi-scale 3D-convolutional neural network for hyperspectral image classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 307–316, 2022, doi: 10.11591/ijeecs.v25.i1.pp307-316.
- [15] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, pp. 1–21, 2021, doi: <https://doi.org/10.1016/j.ymssp.2020.107398>.
- [16] J. Yang, T. Qian, F. Zhang, and S. U. Khan, "Real-Time Facial Expression Recognition Based on Edge Computing," *IEEE Access*, vol. 9, pp. 76178–76190, 2021, doi:

- 10.1109/ACCESS.2021.3082641.
- [17] E. Gamess and S. Hernandez, "Performance Evaluation of Different Raspberry Pi Models for a Broad Spectrum of Interests," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 2, pp. 819–829, 2022, doi: <https://dx.doi.org/10.14569/IJACSA.2022.0130295>.
- [18] J. W. Jolles, "Methods Ecol Evol - 2021 - Jolles - Broad-scale applications of the Raspberry Pi A review and guide for biologists.pdf," *Methods Ecol. Evol.*, vol. 12, no. 9, pp. 1562–1579, 2021.
- [19] A. Magdi, N. A. Lashin, and O. El-komy, "Robust color image watermarking using multi-core Raspberry pi cluster," *Multimed. Tools Appl.*, pp. 17185–17204, 2022.
- [20] A. S. Nugroho, R. Umar, and A. Fadlil, "Klasifikasi Botol Plastik Menggunakan Multiclass Support Vector Machine," *J. Khatulistiwa Inform.*, vol. 9, no. 2, pp. 79–85, 2021, doi: 10.31294/jki.v9i2.11058.
- [21] S. B. Sapin, B. A. Alibudbud, P. B. Molleno, M. B. Veluz, and J. R. Asor, "Intelligent aquaculture system for pisciculture simulation using deep learning algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 561–568, 2022, doi: 10.11591/ijeecs.v25.i1.pp561-568.
- [22] A. E. Minarno, M. H. C. Mandiri, Y. Azhar, F. Bimantoro, H. A. Nugroho, and Z. Ibrahim, "Classification of Diabetic Retinopathy Disease Using Convolutional Neural Network," *Int. J. Informatics Vis.*, vol. 6, no. 1, pp. 12–18, 2022, doi: 10.30630/joiv.6.1.857
- [23] S. M. Abas, A. M. Abdulazeez, and D. Q. Zeebaree, "A YOLO and convolutional neural network for the detection and classification of leukocytes in leukemia," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 200–213, 2022, doi: 10.11591/ijeecs.v25.i1.pp200-213.
- [24] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using VGG19: Caltech-101 image data set," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2021, doi: 10.1007/s12652-021-03488-z.
- [25] S. Patel, R. Patel, N. Ganatra, and A. Patel, "Spatial Feature Fusion for Biomedical Image Classification based on Ensemble Deep CNN and Transfer Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, pp. 153–159, 2022.
- [26] J. Wang, X. Jing, Z. Yan, Y. Fu, W. Pedrycz, and L. T. Yang, "A Survey on Trust Evaluation Based on Machine Learning," *ACM Comput. Surv.*, vol. 53, no. 5, 2020.
- [27] A. Huc, J. Salej, and M. Trebar, "Analysis of Machine Learning Algorithms for Anomaly Detection on Edge Devices," *Sensors*, vol. 21, no. 14, pp. 1–22, 2021.
- [28] N. Bartwal, "Celeb Faces Attributes Dataset (CelebA)," Kaggle. Accessed: Jul. 20, 2025. [Online]. Available: <https://www.kaggle.com/datasets/nikhilbartwal001/celeba>
- [29] M. S. Sayeed, P. P. Min, and M. A. Bari, "Deep Learning Based Gait Recognition Using Convolutional Neural Network in the COVID-19 Pandemic," *Emerg. Sci. J.*, vol. 6, no. 5, pp. 1086–1099, 2022, doi: 10.28991/ESJ-2022-06-05-012.