BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

Analisis Perbandingan Metode Convolutional Neural Network (CNN) untuk Deteksi Warna pada Objek

Dila Ayu Prastita*, Andika Setiawan, Ilham Firman Ashari

Fakultas Teknologi Industri, Program Studi Teknik Informatika, Institut Teknologi Sumatera, Lampung Selatan, Indonesia
Email: [1,*]dila.121140075@student.itera.ac.id, [2]andika.setiawan@if.itera.ac.id, [3]firman.ashari@if.itera.ac.id
Email Penulis Korespondensi: dila.121140075@student.itera.ac.id

Abstrak−Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa tiga arsitektur *Convolutional Neural Network* (CNN), yaitu VGG16, Xception, dan NASNet Mobile, dalam mendeteksi warna pada objek. Permasalahan utama dalam penelitian ini adalah menentukan arsitektur dengan kombinasi *hyperparameter* yang paling efektif dan efisien untuk mendeteksi warna pada objek. Proses penelitian mencakup identifikasi masalah, pengumpulan dataset warna objek, *preprocessing* gambar, pelatihan tiga model CNN (VGG16, Xception, dan NASNet Mobile), serta evaluasi performa menggunakan metrik akurasi, *precision*, *recall*, dan *f1-score*. Selain itu, dilakukan analisis komparatif terhadap performa tiap model berdasarkan kombinasi *hyperparameter* yang digunakan, seperti *optimizer*, *batch size*, dan *learning rate*. Analisis juga mencakup evaluasi efisiensi komputasi dengan mengukur waktu pelatihan dan waktu prediksi masing-masing model, serta mengkaji hubungan antara kompleksitas arsitektur dan performa klasifikasi. Hasil analisis digunakan untuk menentukan model yang paling optimal dan layak diimplementasikan dalam sistem deteksi warna pada objek. Hasil pengujian menunjukkan bahwa NASNet Mobile memberikan performa terbaik dengan akurasi sebesar 88% dan waktu prediksi 2 menit 22 detik untuk 2904 gambar. Model Xception menghasilkan akurasi 86% dengan waktu prediksi 4 menit 22 detik, sementara VGG16 mencatat akurasi 90% dengan waktu prediksi 10 menit 9 detik.

Kata Kunci: CNN; Deteksi Warna; VGG16; Xception; NASNet Mobile

Abstract−This research aims to evaluate and compare the performance of three Convolutional Neural Network (CNN) architectures, namely VGG16, Xception, and NASNet Mobile, in detecting colors on objects. The main problem in this research is to determine the architecture with the most effective and efficient combination of hyperparameters to detect colors on objects. The research process includes problem identification, object color dataset collection, image preprocessing, training of three CNN models (VGG16, Xception, and NASNet Mobile), and performance evaluation using accuracy, precision, recall, and f1-score metrics. In addition, a comparative analysis of the performance of each model based on the combination of hyperparameters used, such as optimizer, batch size, and learning rate. The analysis also includes evaluating computational efficiency by measuring the training time and prediction time of each model, as well as examining the relationship between architectural complexity and classification performance. The results of the analysis are used to determine the most optimal model that is feasible to implement in an object color detection system. The test results show that NASNet Mobile provides the best performance with an accuracy of 88% and a prediction time of 2 minutes 22 seconds for 2904 images. The Xception model produced an accuracy of 86% with a prediction time of 4 minutes 22 seconds, while VGG16 recorded an accuracy of 90% with a prediction time of 10 minutes 9 seconds.

Keywords: CNN; Color Detection; VGG16; Xception; NASNet Mobile

1. PENDAHULUAN

Deteksi warna merupakan salah satu komponen penting dalam berbagai aplikasi *computer vision*, seperti *object detection*, klasifikasi objek, dan pengenalan pola [1]. Seiring dengan kemajuan teknologi, inovasi dalam pengolahan citra digital terus berkembang dan memainkan peran penting dalam proses ekstraksi informasi visual dari gambar. Pengolahan citra digital memungkinkan sistem untuk mengidentifikasi dan menganalisis fitur visual, termasuk warna, dari suatu objek secara lebih efisien. Kemampuan ini sangat penting dalam mendukung pengambilan keputusan berbasis visual yang cepat dan akurat di berbagai bidang, mulai dari industri hingga otomasi sistem cerdas [2]. Deteksi warna juga berpotensi memberikan manfaat bagi penyandang buta warna parsial, seperti *deuteranomaly* dan *tritanomali*, yang umumnya mengalami kesulitan dalam membedakan warna-warna dasar seperti merah, hijau, biru, dan kuning [3]. Jenis gangguan ini lebih umum terjadi dibandingkan jenis buta warna total [4]. Oleh karena itu, penelitian ini memfokuskan deteksi pada empat warna tersebut, dengan harapan hasilnya dapat memberikan rekomendasi model terbaik untuk digunakan dalam pengembangan sistem pendeteksi warna yang lebih inklusif dan adaptif terhadap kebutuhan penderita buta warna parsial.

Seiring dengan berkembangnya teknologi, berbagai pendekatan berbasis kecerdasan buatan telah diusulkan untuk mengatasi permasalahan ini. Salah satu solusi yang potensial untuk mendeteksi warna secara akurat dalam berbagai kondisi adalah *Deep Learning* (DL) [5]. Di antara metode DL, *Convolutional Neural Network* (CNN) menjadi yang paling menonjol dalam pengenalan citra [5]. CNN terbukti unggul dalam menangkap pola visual kompleks, termasuk klasifikasi warna pada gambar [6]. CNN mampu menerima input berupa gambar, mengenali objek, serta memprediksi gambar dengan akurat [6]. CNN memiliki keunggulan dalam hal akurasi tinggi dan efisiensi komputasi. Selain itu, CNN juga melampaui metode *machine learning* (ML) konvensional yang masih bergantung pada ekstraksi fitur manual dan kurang efektif dalam kondisi lingkungan yang dinamis [7]. CNN mampu mengekstraksi fitur secara hierarkis melalui lapisan konvolusi, menjadikannya unggul dalam mengenali pola kompleks [7].

Dari berbagai pendekatan yang tersedia, CNN dipilih dalam penelitian ini karena kemampuannya dalam mendeteksi fitur visual kompleks. Beberapa arsitektur CNN yang umum digunakan meliputi VGG16, Xception, dan NASNet Mobile [8]. VGG16 memiliki arsitektur mendalam yang akurat dalam mengenali pola visual, namun memerlukan waktu perhitungan yang lebih lama [8]. Xception menawarkan efisiensi waktu pelatihan dan akurasi tinggi

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

melalui arsitektur yang lebih kompleks namun ringan [9]. NASNet Mobile dirancang khusus untuk perangkat *mobile*, dengan arsitektur *scalable* yang dioptimalkan melalui pembelajaran penguatan [10]. Ketiga model ini dipilih karena memiliki kemampuan mengekstraksi fitur gambar kompleks dan mendukung klasifikasi visual secara akurat. Pemilihan ketiga arsitektur ini didasarkan pada pertimbangan performa akurasi dan kecepatan proses komputasi, dua aspek utama yang menjadi fokus dalam penelitian ini. Perbandingan dilakukan untuk mengevaluasi sejauh mana masing-masing model mampu menyeimbangkan antara ketepatan dalam deteksi warna dan efisiensi waktu, sehingga dapat digunakan sebagai solusi praktis baik untuk *platform mobile* maupun *desktop*.

Evaluasi kinerja model deteksi warna berbasis DL menggunakan metrik *accuracy*, *precision*, *recall*, dan *f1-score*. *Accuracy* mengukur proporsi prediksi benar dari total prediksi, *precision* menilai ketepatan model dalam mendeteksi warna yang benar, *recall* mengukur kemampuan model menangkap semua warna yang benar, dan *f1-score* menyeimbangkan *precision* dan *recall* [11]. Metrik ini penting untuk memastikan model tidak hanya akurat, tetapi juga praktis dan konsisten dalam mendeteksi warna pada objek.

Beberapa penelitian sebelumnya telah mengaplikasikan berbagai arsitektur *Convolutional Neural Network* (CNN) untuk klasifikasi citra dengan hasil yang menjanjikan. Farid et al. (2023) mengevaluasi model CNN dengan arsitektur Xception untuk klasifikasi citra SIBI dan berhasil mencapai *f1-score* sebesar 99,57% serta akurasi yang hampir sempurna, yaitu 99,57%. Sedangkan VGG16 memiliki performa terburuk, yaitu dengan *f1-score* 42,13% dan akurasinya sebesar 42,13% [8]. Selain itu, Rahman et al. (2024) membandingkan beberapa arsitektur CNN untuk klasifikasi gambar dermoskopik, dimana NASNet Mobile menunjukkan performa tinggi dengan akurasi mencapai 85,62% [12]. Namun, sebagian besar penelitian sebelumnya belum mengeksplorasi secara komprehensif efisiensi model serta pengaruh variasi *hyperparameter* dalam konteks tugas deteksi warna objek, sehingga hal ini menjadi fokus utama dalam penelitian ini.

Penelitian ini bertujuan untuk membandingkan kinerja tiga arsitektur *Convolutional Neural Network* (CNN), yaitu VGG16, Xception, dan NASNet Mobile, dalam mendeteksi warna merah, hijau, biru, dan kuning pada objek. Setiap model diuji dengan delapan variasi pengaturan *hyperparameter* untuk mengevaluasi performa dan ketepatan deteksi warna pada kondisi yang berbeda. Melalui perbandingan ini, penelitian diharapkan dapat memberikan rekomendasi arsitektur CNN yang paling optimal dan menjadi acuan bagi pengembangan sistem pendeteksi warna yang lebih efektif, khususnya untuk mendukung kebutuhan pengguna dengan gangguan buta warna parsial.

2. METODOLOGI PENELITIAN

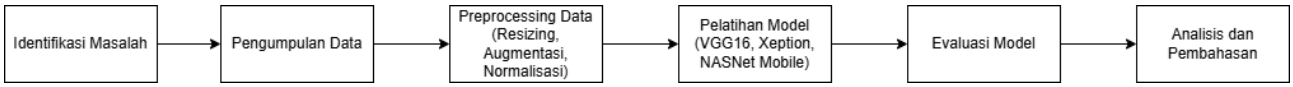
2.1 Kerangka Dasar Penelitian

Penelitian ini merupakan penelitian kuantitatif eksperimental yang bertujuan membandingkan performa tiga arsitektur CNN (VGG16, Xception, dan NASNet Mobile) dalam mendeteksi warna pada objek. Penelitian dilakukan di Institut Teknologi Sumatera, menggunakan lebih dari 14.532 citra objek dari dataset sekunder dan hasil augmentasi. Penelitian tidak melibatkan responden manusia, melainkan data citra dengan empat kelas warna utama (merah, hijau, biru, kuning). Hipotesis nol (H_0) menyatakan tidak ada perbedaan performa antara model VGG16, Xception, dan NASNet Mobile, sedangkan hipotesis alternatif (H_1) menyatakan ada perbedaan yang signifikan.

Variabel bebas adalah jenis arsitektur CNN dan kombinasi *hyperparameter*, sedangkan variabel terikat adalah performa model yang diukur melalui akurasi, *precision*, *recall*, *f1-score*, serta waktu komputasi. Teknik analisis menggunakan *confusion matrix* dan evaluasi metrik performa tiap model. Kerangka pemikiran penelitian didasarkan pada kebutuhan sistem deteksi warna yang akurat dan efisien untuk mendukung pengguna dengan keterbatasan penglihatan warna. Penelitian ini memberikan rekomendasi arsitektur CNN terbaik berdasarkan keseimbangan akurasi dan efisiensi waktu.

2.2 Tahapan Penelitian

Alur yang digunakan dalam pengembangan model deteksi warna pada objek menggunakan metode *Convolutional Neural Network* (CNN) ditunjukkan pada Gambar 2.1. Alur ini terdiri dari beberapa tahapan utama, dimulai dari identifikasi masalah, pengumpulan dan *preprocessing* data, pelatihan model CNN menggunakan beberapa arsitektur, evaluasi kinerja model, hingga analisis dan pembahasan.



Gambar 2.1. Tahapan Penelitian

Gambar 2.1 menggambarkan alur penelitian yang telah dilaksanakan. Penjabaran masing-masing tahapannya diuraikan sebagai berikut.

a. Identifikasi Masalah
   Langkah awal penelitian ini adalah identifikasi masalah melalui studi literatur terkait metode deteksi warna objek menggunakan CNN. Ditemukan bahwa arsitektur seperti VGG16, Xception, dan NASNet Mobile memiliki performa yang bervariasi tergantung konfigurasi. Namun, belum banyak studi yang membandingkan performa ketiganya, khususnya dalam tugas deteksi warna. Selain itu, pengaruh hyperparameter seperti *batch size*, *optimizer*, dan *learning*

ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

rate juga belum dikaji secara mendalam. Oleh karena itu, penelitian ini berfokus pada analisis perbandingan arsitektur CNN dan variasi hyperparameternya untuk menemukan model paling optimal.

b. Pengumpulan Data
Penelitian ini menggunakan kombinasi dataset sekunder untuk pelatihan dan evaluasi model CNN dalam mendeteksi warna pada objek. Dataset sekunder diperoleh dari https://www.kaggle.com/datasets/trushraut18/colour-classification untuk kelas merah, hijau, biru, serta dua dataset tambahan untuk melengkapi kelas warna kuning yang diperoleh dari https://www.kaggle.com/datasets/imoore/6000-store-items-images-classified-by-color?resource=download dan https://www.kaggle.com/datasets/brasarkaya/vehicle-color-dataset.

c. *Preprocessing* Data
Setelah dataset disiapkan, tahap *preprocessing* gambar dilakukan untuk memastikan data siap digunakan dalam pelatihan model. Langkah-langkah *preprocessing* meliputi:
1. *Resizing* gambar : gambar diubah ke ukuran 128×128 piksel untuk memastikan keseragaman ukuran.
2. Augmentasi data : Untuk meningkatkan keragaman data tanpa menambah data baru [13], dilakukan augmentasi berupa rotasi 40°, flipping horizontal, dan flipping vertikal. Setiap gambar asli diaugmentasi sebanyak tiga kali menggunakan ketiga teknik tersebut, sehingga menghasilkan empat variasi per gambar (termasuk gambar asli). Setelah proses augmentasi, total jumlah data menjadi 14.532 gambar.
3. Normalisasi piksel : mengubah nilai piksel dalam rentang [0, 1] untuk memudahkan model dalam belajar dan mencapai konvergensi lebih cepat [14].

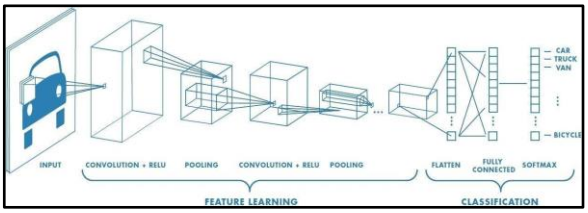
d. Pelatihan Model
Penelitian ini menggunakan tiga arsitektur CNN, yaitu VGG16, Xception, dan NASNet Mobile, untuk tugas klasifikasi warna objek. Masing-masing model diuji dengan delapan kombinasi *hyperparameter*, mencakup *optimizer* (Adam dan SGD), *learning rate* (0,001 dan 0,0001), dan *batch size* (32 dan 64), guna mengevaluasi pengaruhnya terhadap performa model. Dataset berjumlah 14.532 gambar setelah proses augmentasi, dibagi menjadi 80% data latih dan 20% data uji, tanpa validasi silang. Pelatihan dilakukan selama 10 epoch.

e. Evaluasi Model
Setelah pelatihan, ketiga model diuji menggunakan akurasi, *precision*, *recall*, dan *f1-score*, serta dibandingkan berdasarkan variasi *hyperparameter*. Evaluasi juga mencakup waktu pelatihan dan prediksi untuk menilai efisiensi dan kecepatan respon masing-masing model.

f. Analisis dan Pembahasan
Setelah pengujian, dilakukan analisis performa ketiga model dan delapan variasi *hyperparameter* untuk menentukan model paling efektif dalam klasifikasi warna. Analisis mencakup pengaruh *hyperparameter*, perbandingan metrik evaluasi, serta efisiensi waktu komputasi. Hasilnya digunakan untuk mengidentifikasi kelebihan tiap model dan memberikan rekomendasi model terbaik serta acuan bagi penelitian selanjutnya.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara [15]. CNN banyak digunakan dalam analisis gambar visual untuk mendeteksi dan mengenali objek pada gambar, yang merupakan vektor berdimensi tinggi dan memerlukan banyak parameter untuk mendefinisikan jaringan. Secara umum, CNN memiliki prinsip yang mirip dengan *neural network* pada umumnya, terdiri dari *neuron* yang memiliki bobot (*weight*), bias, dan fungsi aktivasi (*activation function*) [6].



Gambar 2.2. Arsitektur *Convolutional Neural Network* (CNN)

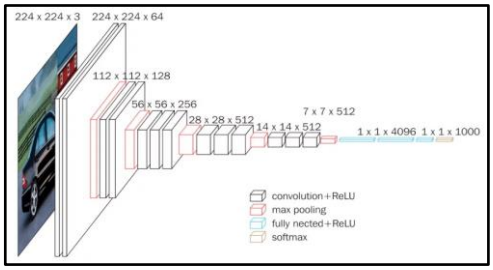
Pada CNN, lapisan-lapisan (*layers*) dibagi menjadi dua bagian utama berdasarkan fungsinya, yaitu lapisan ekstraksi fitur dan lapisan klasifikasi. Lapisan ekstraksi fitur bertugas untuk mengambil fitur-fitur penting dari citra input, sementara lapisan klasifikasi bertugas mengklasifikasikan citra berdasarkan fitur-fitur yang dihasilkan dari lapisan ekstraksi. Lapisan ekstraksi fitur mencakup *convolutional layer*, *pooling layer*, dan *rectified linear unit* (ReLU). Sedangkan lapisan klasifikasi terdiri dari *flatten*, *fully connected layer*, dan *softmax* [16].

2.4 Arsitektur VGG16

VGG16 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang paling standar, populer, dan sering digunakan untuk tugas-tugas pengenalan gambar. VGG16 adalah model yang kompleks dengan waktu perhitungan yang lambat, namun dapat menghasilkan hasil klasifikasi yang akurat, menjadikannya pilihan yang sangat baik [8]. VGG16 memiliki total 16 lapisan, di mana lapisan-lapisan tersebut diatur dalam blok-blok konvolusi. Setiap blok terdiri dari

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

beberapa lapisan konvolusi dengan filter berukuran 3x3, diikuti oleh lapisan aktivasi ReLU, dan diakhiri dengan lapisan MaxPooling yang bertujuan untuk mengurangi dimensi spasi peta fitur [17]. Arsitektur VGG16 dapat dilihat pada Gambar 2.3.

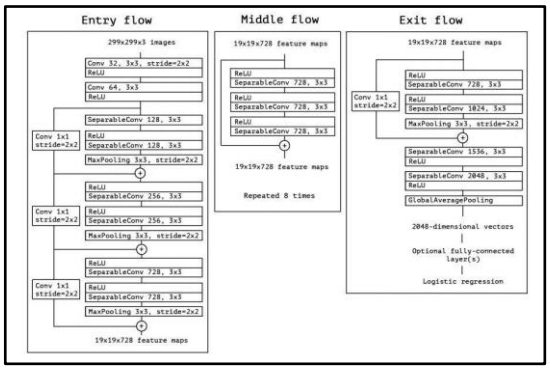


Gambar 2.3. Arsitektur VGG16

Secara keseluruhan, VGG16 memiliki 13 lapisan konvolusi dan 3 lapisan *Fully Connected* (FC). Blok-blok konvolusi ini secara bertahap mengekstraksi fitur dari gambar dengan berbagai tingkat kompleksitas. Penggunaan lapisan konvolusi berturut-turut memungkinkan VGG16 mempelajari representasi hierarkis gambar, mulai dari deteksi fitur dasar seperti garis tepi hingga fitur yang lebih kompleks seperti tekstur dan bentuk [17].

2.5 Arsitektur Xception

Model Xception memiliki arsitektur berupa tumpukan linear yang terdiri dari lapisan *depthwise separable convolution*, yang dilengkapi dengan *residual connections*. Arsitektur Xception terbagi menjadi tiga bagian utama, yaitu *entry flow*, *middle flow*, dan *exit flow* [9]. Arsitektur Xception dapat dilihat pada Gambar 2.4.

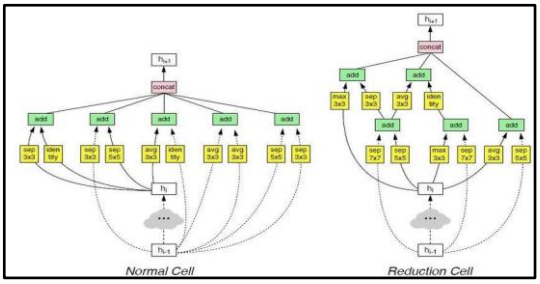


Gambar 2.4. Arsitektur Xception

Pada bagian *entry flow*, citra *input* diterima untuk diproses. Kemudian, *middle flow* menggunakan *Separable Convolution*, variasi dari *Depthwise Separable Convolution*, dengan fungsi aktivasi ReLU yang diterapkan pada setiap lapisan konvolusi dan *fully connected* untuk memperdalam jaringan dan melakukan ekstraksi fitur dengan kompleksitas tinggi. Bagian terakhir dari arsitektur Xception adalah *exit flow* yang bertanggung jawab untuk menyimpulkan informasi dari fitur yang telah dipelajari. Setelah *middle flow* selesai mengekstrak fitur tingkat tinggi, *exit flow* menerapkan lapisan *convolutional* terakhir serta melakukan *pooling* untuk mengurangi dimensi akhir peta fitur [9].

2.6 Arsitektur NASNet Mobile

NASNet Mobile adalah arsitektur CNN ringan yang dirancang untuk perangkat *mobile* menggunakan pendekatan *Neural Architecture Search* (NAS), yaitu pencarian otomatis untuk menemukan blok konvolusional paling optimal [10]. Model ini dilatih menggunakan dataset ImageNet dan dibangun dari dua jenis sel: sel normal dan sel reduksi, yang disusun untuk menghasilkan kinerja klasifikasi tinggi dengan beban komputasi rendah [12]. Arsitektur NASNet Mobile dapat dilihat pada Gambar 2.5.



Gambar 2.5. Arsitektur Xception

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

NASNet terdiri dari dua komponen utama: *Normal Cell* dan *Reduction Cell*. *Normal Cell* menjaga ukuran peta fitur, sedangkan *Reduction Cell* menguranginya dalam dimensi spasial. NASNet awalnya dilatih pada dataset kecil, lalu arsitekturnya dipindahkan ke dataset besar untuk meningkatkan performa. Teknik *Scheduled Droppath* digunakan untuk menonaktifkan jalur secara bertahap selama pelatihan, meningkatkan stabilitas dan akurasi model [18].

2.7 *Confusion Matrix*

Confusion matrix digunakan untuk mengevaluasi hasil klasifikasi model dengan menghitung *accuracy*, *precision*, *recall*, dan *f1-score* yang mencerminkan kemampuan model dalam mengenali warna objek secara akurat. Confusion matrix umum digunakan dalam pengolahan citra dan pembelajaran mesin untuk memberikan gam aran menyeluruh terkait performa model [19]. Rumus perhitungan *Confusion Matrix accuracy*, *precision*, *recall*, dan *f1-Score* untuk setiap variasi arsitektur CNN dapat dilihat secara berturut turut pada persamaan (1), (2), (3), dan (4). TP adalah jumlah *True Positive*, TN adalah jumlah *True Negative*, FP adalah jumlah *False Positive*, dan FN adalah jumlah *False Negative* [20].

a. *Accuracy* mengukur persentase prediksi yang benar dari total prediksi yang dilakukan oleh model.

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

b. *Precision* adalah rasio antara jumlah deteksi benar (*True Positives*) dengan total jumlah prediksi positif (*True Positives* + *False Positives*).

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

c. *Recall* adalah rasio antara jumlah deteksi benar (*True Positives*) dengan total jumlah data positif sebenarnya (*True Positives* + *False Negatives*).

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

d. *F1-Score* adalah rata-rata harmonis dari *Precision* dan *Recall*, yang memberikan gambaran keseimbangan antara keduanya.

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Sebelum dilakukan pelatihan model, tahap *preprocessing* data dilakukan untuk mempersiapkan dataset secara optimal. *Preprocessing* data dimulai dengan mengubah ukuran (*resize*) seluruh citra menjadi 128x128 piksel agar seragam dan sesuai dengan kebutuhan *input* model. Setelah itu dilakukan augmentasi data untuk meningkatkan variasi dataset dan mengurangi *overfitting*. Teknik augmentasi yang digunakan antara lain rotasi, flipping horizontal, dan flipping vertikal. Tabel 3.1 berikut menunjukkan perbandingan jumlah dataset sebelum dan sesudah augmentasi.

Tabel 3.1. Total dataset

	Sebelum Augmentasi		Setelah Augmentasi	
Kelas	Train	Test	Train	Test
Merah	800	200	3200	800
Hijau	800	200	3200	800
Biru	800	200	3200	800
Kuning	507	126	2028	504
Total	2907	726	11628	2904

Selanjutnya, semua citra dinormalisasi dengan membagi nilai piksel dengan 255 untuk mengubah rentang nilai piksel dari 0-255 menjadi 0-1. Setelah tahap *preprocessing* selesai, Pelatihan model dilakukan dengan menerapkan beberapa variasi *hyperparameter*. Model dilatih dengan menggunakan 10 *epoch* dan menggunakan fungsi aktivasi *softmax* pada layer *output* untuk menghasilkan probabilitas dari masing-masing kelas. Masing masing arsitektur CNN seperti VGG16, Xception, dan NASNet Mobile divariasikan menjadi 8 variasi untuk menguji bagaimana perubahan *hyperparameter* mempengaruhi kinerja model dalam mendeteksi warna. Adapun variasi *hyperparameter* dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2. Variasi *hyperparameter*

Variasi	Optimizer	Batch Size	Learning Rate
1	Adam	32	0,001
2	Adam	32	0,0001
3	Adam	64	0,001
4	Adam	64	0,0001

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

Variasi	Optimizer	Batch Size	Learning Rate
5	SGD	32	0,001
6	SGD	32	0,0001
7	SGD	64	0,001
8	SGD	64	0,0001

Berdasarkan variasi *hyperparameter* yang telah ditentukan pada Tabel 3.2, dilakukan evaluasi performa model VGG16 dengan menggunakan *confusion matrix* yang disajikan pada Tabel 3.3. Tabel 3.3 menampilkan hasil klasifikasi model terhadap kombinasi *optimizer*, *batch size*, dan *learning rate* yang berbeda-beda.

Tabel 3.3. Hasil variasi *hyperparameter* VGG16

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Confusion Matrix*				Metrik Evaluasi			
				TP	TN	FP	FN	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	Adam	32	0.001	2586	8394	318	318	89 %	89 %	90 %	90 %
2			0.0001	2589	8397	315	315	89 %	89 %	90 %	90 %
3		64	0.001	2603	8411	301	301	90 %	90 %	90 %	90 %
4			0.0001	2573	8381	331	331	89 %	89 %	89 %	89 %
5	SGD	32	0.001	2262	8070	642	642	78 %	78 %	79 %	79 %
6			0.0001	1705	7513	1199	1199	59 %	60 %	61 %	60 %
7		64	0.001	2169	7977	735	735	75 %	75 %	76 %	76 %
8			0.0001	1383	7191	1521	1521	48 %	53 %	46 %	48 %

Berdasarkan Tabel 3.3, performa terbaik model VGG16 diperoleh pada variasi ke-3 dengan kombinasi *optimizer* Adam, *batch size* 64, dan *learning rate* 0.001. Kombinasi ini menghasilkan *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 90 %, serta nilai TP dan TN tertinggi (2603 dan 8411), menunjukkan kemampuan klasifikasi yang konsisten. Sebaliknya, performa terburuk terdapat pada variasi ke-8 (SGD, *batch size* 64, *learning rate* 0.0001) dengan *accuracy* 48 %, serta nilai *precision*, *recall*, dan *F1-score* yang rendah (masing-masing 53 %, 46 %, dan 48 %). Hal ini disebabkan oleh kombinasi *hyperparameter* yang kurang optimal, terutama *learning rate* yang terlalu kecil dan keterbatasan SGD dalam konvergensi. Secara umum, *optimizer* Adam unggul dibandingkan SGD di seluruh kombinasi. *Batch size* besar (64) juga efektif jika didukung oleh *learning rate* yang sesuai.

Selain evaluasi metrik, efisiensi waktu komputasi juga penting, mencakup waktu pelatihan (*training time*) dan inferensi (*prediction time*). Oleh karena itu, Tabel 3.4 disajikan untuk membandingkan aspek waktu dari tiap variasi model.

Tabel 3.4. *Computational time* untuk model VGG16

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Computational Time*	
				Training Time	*Prediction Time*
1	Adam	32	0.001	03:06:00	00:09:22
2			0.0001	03:22:08	00:09:58
3		64	0.001	03:23:44	00:10:09
4			0.0001	03:23:10	00:10:03
5	SGD	32	0.001	03:25:24	00:08:58
6			0.0001	03:21:22	00:10:22
7		64	0.001	03:19:24	00:09:52
8			0.0001	03:24:50	00:09:59

Berdasarkan Tabel 3.4, seluruh variasi model VGG16 memiliki waktu pelatihan yang relatif mirip, yaitu antara 3 hingga 3 jam 25 menit. Variasi ke-3 (Adam, *batch size* 64, *learning rate* 0.001) mencatat *training time* 3 jam 23 menit 44 detik dan *prediction time* 10 menit 9 detik. Meski bukan yang tercepat, variasi ini tetap unggul karena memberikan akurasi tertinggi. *Optimizer* SGD memang lebih cepat dalam prediksi, seperti pada variasi ke-5 (8 menit 58 detik), namun hasil metriknya jauh lebih rendah dibandingkan Adam. Oleh karena itu, efisiensi komputasi perlu diimbangi dengan kualitas prediksi. VGG16 variasi ke-3 merupakan kombinasi *hyperparameter* terbaik karena menghasilkan performa tinggi dan efisiensi waktu yang tetap layak.

Setelah mengevaluasi performa model VGG16, selanjutnya dilakukan pengujian pada model Xception yang dapat dilihat pada Tabel 3.5.

Tabel 3.5. Hasil variasi *hyperparameter* Xception

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Confusion Matrix*				Metrik Evaluasi			
				TP	TN	FP	FN	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	Adam	32	0.001	2492	8300	412	412	86 %	86 %	87 %	86 %

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Confusion Matrix*				Metrik Evaluasi			
				TP	TN	FP	FN	*Accuracy*	*Precision*	*Recall*	*F1-Score*
2			0.0001	2491	8298	414	414	86 %	86 %	87 %	86 %
3		64	0.001	2504	8312	400	400	86 %	86 %	87 %	87 %
4			0.0001	2477	8285	427	427	85 %	86 %	86 %	86 %
5	SGD	32	0.001	2269	8077	635	635	78 %	79 %	80 %	79 %
6			0.0001	1456	7264	1448	1448	50 %	52 %	51 %	51 %
7		64	0.001	2148	7956	756	756	74 %	74 %	76 %	75 %
8			0.0001	1015	6823	1889	1889	35 %	35 %	37 %	35 %

Berdasarkan Tabel 3.5, variasi 1-4 dengan *optimizer* Adam menunjukkan performa yang lebih stabil dan unggul dibandingkan variasi 5-8 yang menggunakan SGD, ditinjau dari nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Di antaranya, variasi ke-3 (Adam, *batch size* 64, *learning rate* 0.001) memberikan hasil terbaik dengan *accuracy* 86 %, *precision* 86 %, *recall* 87 %, dan *f1-score* 87 %. Performa ini didukung oleh nilai TP dan TN yang tinggi (2504 dan 8312), serta FP dan FN yang rendah (400), menunjukkan kemampuan model dalam mengklasifikasikan data secara akurat dan seimbang. Sebaliknya, variasi ke-8 (SGD, *batch size* 64, *learning rate* 0.0001) mencatat performa terburuk dengan *accuracy* dan *F1-score* hanya 35 %. Nilai TP dan TN yang rendah (1015 dan 6823), serta FP dan FN yang tinggi yaitu 1889, mengindikasikan bahwa model gagal belajar secara optimal. Kemungkinan besar hal ini disebabkan oleh *learning rate* yang terlalu kecil dan karakteristik SGD yang membutuhkan konfigurasi lebih tepat. Selanjutnya, Tabel 3.6 menyajikan perbandingan waktu komputasi dari masing-masing variasi model Xception.

Tabel 3.6. *Computational time* untuk model Xception

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Computational Time*	
				Training Time	*Prediction Time*
1	Adam	32	0.001	01:49:08	00:03:24
2			0.0001	01:11:09	00:03:22
3		64	0.001	01:13:55	00:04:22
4			0.0001	01:36:37	00:03:45
5	SGD	32	0.001	01:12:04	00:03:37
6			0.0001	01:11:22	00:03:34
7		64	0.001	01:15:27	00:03:46
8			0.0001	02:11:10	00:04:07

Meskipun variasi ke-3 memberikan performa evaluasi terbaik pada Tabel 3.5, ia bukan yang paling efisien secara waktu. *Training time*-nya 1 jam 13 menit 55 detik dan *prediction time* 4 menit 22 detik, lebih lama dibandingkan variasi lainnya. Namun, variasi ke-3 tetap layak dipilih karena akurasi dan skor evaluasinya tertinggi, sementara selisih waktu komputasi masih dapat ditoleransi.

Setelah mengevaluasi performa model Xception, analisis selanjutnya difokuskan pada model NASNet yang hasilnya dapat dilihat pada Tabel 3.7.

Tabel 3.7. Hasil variasi *hyperparameter* NASNet Mobile

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Confusion Matrix*				Metrik Evaluasi			
				TP	TN	FP	FN	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	Adam	32	0.001	2547	8355	357	357	88 %	88 %	98 %	88 %
2			0.0001	2504	8312	400	400	86 %	87 %	87 %	87 %
3		64	0.001	2532	8340	372	372	87 %	87 %	88 %	88 %
4			0.0001	2480	8288	424	424	85 %	86 %	86 %	86 %
5	SGD	32	0.001	2264	8072	640	640	78 %	78 %	80 %	79 %
6			0.0001	1349	7157	1555	1555	46 %	45 %	48 %	46 %
7		64	0.001	2052	7860	852	852	71 %	71 %	72 %	71 %
8			0.0001	1049	6857	1855	1855	36 %	36 %	38 %	35 %

Berdasarkan Tabel 3.7, performa terbaik model NASNet Mobile diperoleh pada variasi pertama (Adam, *batch size* 32, *learning rate* 0.001) dengan akurasi, *precision*, dan *f1-score* masing-masing sebesar 88 % dengan *recall* sebesar 98 %, serta nilai TP dan TN tertinggi di antara semua variasi. Ini menunjukkan kemampuan klasifikasi model yang konsisten untuk kelas positif dan negatif. Nilai *recall* yang tinggi yaitu 98 % menunjukkan model sangat andal dalam mendeteksi kelas positif, meski *precision* lebih rendah yaitu 88 % akibat *trade-off* dengan *false positive*. *Trade-off* ini menguntungkan karena penelitian memprioritaskan *recall* tinggi untuk meminimalkan kesalahan pendeteksian warna. Sebaliknya, variasi ke-8 (SGD, *batch size* 64, *learning rate* 0.0001) menunjukkan performa terburuk, dengan akurasi hanya 36 % dan tingginya kesalahan klasifikasi (FP dan FN yaitu 1855). Secara keseluruhan, *optimizer* Adam terbukti lebih efektif

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 821-830
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.617

dibandingkan SGD, dan *learning rate* terlalu kecil cenderung menurunkan performa, terutama saat dikombinasikan dengan SGD. Analisis efisiensi waktu komputasi untuk model ini disajikan pada Tabel 3.8.

Tabel 3.8. *Computational time* untuk model NASNet Mobile

Variasi	*Optimizer*	*Batch Size*	*Learning Rate*	*Computational Time*	
				Training Time	*Prediction Time*
1	Adam	32	0.001	01:40:54	00:02:22
2			0.0001	00:31:48	00:01:37
3		64	0.001	00:31:41	00:01:44
4			0.0001	00:33:02	00:01:42
5	SGD	32	0.001	00:30:51	00:01:41
6			0.0001	00:29:43	00:01:39
7		64	0.001	00:31:33	00:01:44
8			0.0001	00:31:55	00:02:22

Berdasarkan Tabel 3.8, variasi pertama (Adam, *batch size* 32, *learning rate* 0.001) mencatat waktu pelatihan terlama, yakni 1 jam 40 menit 54 detik, namun memberikan performa terbaik berdasarkan Tabel 3.7. Sebaliknya, variasi ke-6 (SGD, *batch size* 32, *learning rate* 0.0001) memiliki waktu pelatihan tercepat hanya 29 menit 43 detik dan prediksi 1 menit 39 detik, tetapi performanya jauh lebih rendah. Hal ini menunjukkan bahwa efisiensi waktu komputasi tidak selalu sejalan dengan kualitas prediksi, sehingga pemilihan parameter perlu mempertimbangkan keseimbangan antara performa dan efisiensi.

3.2 Analisis Hasil

Dilakukan perbandingan variasi terbaik dari VGG16, Xception, dan NASNet Mobile berdasarkan kombinasi *hyperparameter* dengan performa evaluasi tertinggi. Hasilnya ditampilkan pada Tabel 3.9, mencakup metrik akurasi, *precision*, *recall*, *F1-score*, serta waktu pelatihan dan prediksi.

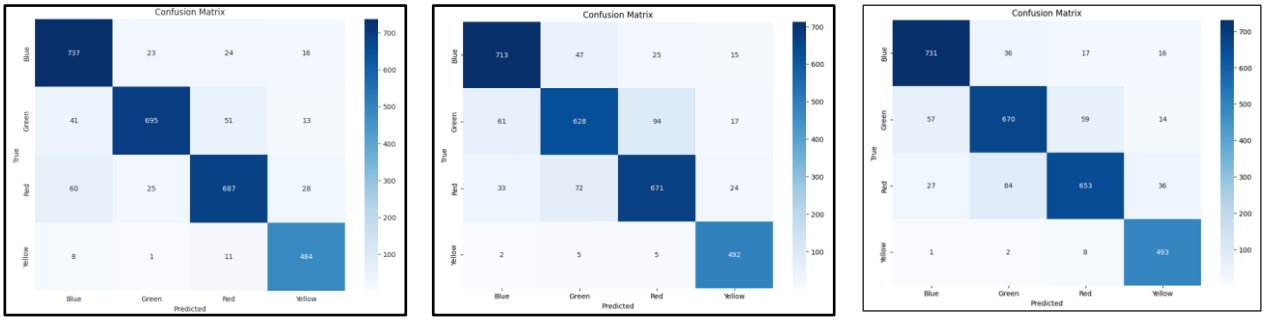
Tabel 3.9. Hasil untuk variasi terbaik dari masing-maisng arsitektur

Model	*Optimizer*	*Batch Size*	*Learning Rate*	Metrik Evaluasi				*Computational Time*	
				Accuracy	*Precision*	*Recall*	*F1-Score*	*Training Time*	*Prediction Time*
VGG16	Adam	64	0.001	0.90	0.90	0.90	0.90	03:23:44	00:10:09
Xception	Adam	64	0.001	0.86	0.86	0.87	0.87	01:13:55	00:04:22
NASNet Mobile	Adam	32	0.001	0.88	0.88	0.98	0.88	01:40:54	00:02:22

Berdasarkan hasil pengujian Tabel 3.9, setiap arsitektur CNN menunjukkan karakteristik performa yang berbeda:

a. VGG16 menunjukkan akurasi tertinggi pada variasi ke-3 dengan akurasi dan *f1-score* sebesar 90% menggunakan *optimizer* Adam, *batch size* 64, dan *learning rate* 0.001. Namun, VGG16 memiliki kelemahan pada durasi pelatihan (>3 jam) dan prediksi (>8 menit), serta performa yang menurun drastis saat menggunakan *optimizer* SGD.
b. Xception tampil stabil pada variasi 1-4, dengan performa terbaik pada variasi ke-3 dengan akurasi sebesar 86% dan *f1-score* sebesar 87%. Model ini juga menggunakan *optimizer* Adam, *batch size* 64, dan *learning rate* 0.001, serta memiliki keunggulan efisiensi waktu, dengan waktu pelatihan selama ±1 jam 30 menit dan prediksi selama 3-4 menit.
c. NASNet Mobile menjadi model paling optimal dan efisien, khususnya pada variasi ke-1 dengan akurasi dan *F1-score* sebesar 88%, serta waktu pelatihan 1 jam 40 menit serta prediksi hanya 2 menit. Model ini dinilai potensial untuk diimplementasikan karena sifatnya yang ringan, akurat, dan cepat, sehingga direkomendasikan sebagai pilihan terbaik untuk penelitian selanjutnya dalam pengembangan model deteksi warna pada objek.

Untuk memberikan gambaran yang lebih jelas mengenai performa klasifikasi dari ketiga model tersebut, Gambar 3.1 menampilkan visualisasi *confusion matrix* dari masing-masing model terbaik.



VGG16 Variasi 2 | Xception Variasi 3 | NASNet Mobile Variasi 1

Gambar 3.1. Visualisasi Confusion Matrix Model Terbaik

Adapun beberapa faktor yang memengaruhi tinggi atau rendahnya akurasi dari masing-masing model dan variasi adalah:

a. *Optimizer*: *Optimizer* Adam selalu menunjukkan hasil lebih baik dibandingkan SGD karena mampu menyesuaikan *learning rate* secara adaptif untuk setiap parameter sehingga konvergensi lebih cepat. Sedangkan SGD menggunakan *learning rate* tetap tanpa penyesuaian adaptif, sehingga performanya lebih rendah.
b. *Learning Rate*: *Learning rate* terlalu kecil (0.0001) membuat pelatihan lambat dan tidak mencapai konvergensi optimal, sementara *learning rate* 0.001 lebih stabil dan efektif.
c. *Batch Size*: Hasil pengujian menunjukkan performa model tidak mutlak tergantung *batch size*. *Batch size* 32 sering baik karena pembaruan bobot lebih sering, tapi *batch size* 64 juga optimal pada model seperti VGG16 dan Xception. Pengaruh *batch size* tergantung pada arsitektur model dan *learning rate*.
d. Arsitektur: VGG16 akurasi tertinggi tapi pelatihan dan prediksi paling lama, Xception seimbang antara akurasi dan efisiensi tapi sensitif terhadap variasi citra, NASNet Mobile paling efisien dengan akurasi cukup tinggi dan waktu pelatihan serta prediksi tercepat.

NASNet Mobile memiliki waktu perhitungan tercepat dibandingkan Xception dan VGG16 karena menggunakan pendekatan *Neural Architecture Search* (NAS) yang secara otomatis menemukan arsitektur optimal dengan jumlah parameter dan komputasi minimal. Selain itu, teknik *Scheduled Droppath* membantu mempercepat proses pelatihan dan inferensi. Xception lebih cepat dari VGG16 karena menggunakan *Depthwise Separable Convolution*, yang mengurangi jumlah operasi konvolusi secara signifikan namun tetap menghasilkan akurasi tinggi. Sebaliknya, VGG16 merupakan model yang lebih kompleks karena menggunakan banyak lapisan konvolusi 3×3 berturut-turut dan *fully connected layer* besar, sehingga membutuhkan waktu pelatihan dan prediksi yang lebih lama.

4. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa pemilihan arsitektur CNN sangat memengaruhi performa klasifikasi warna objek, baik dari segi akurasi maupun efisiensi waktu. VGG16 memberikan akurasi *f1-score* tertinggi sebesar 90%, namun memiliki kelemahan pada durasi pelatihan selama 3 jam 23 menit serta waktu prediksi selama 10 menit 9 detik. Selain itu, performanya menurun signifikan saat menggunakan *optimizer* SGD. Sementara itu, Xception menunjukkan keseimbangan antara akurasi dan efisiensi waktu, dengan akurasi sebesar 86% dan *f1-score* sebesar 87%, waktu pelatihan sekitar 1 jam 13 menit, dan waktu prediksi 4 menit 22 detik. Di sisi lain, NASNet Mobile merupakan arsitektur yang paling efisien dan optimal dengan akurasi dan *f1-score* sebesar 88%. NASNet Mobile juga mencatat waktu pelatihan hanya 1 jam 40 menit dan prediksi tercepat yaitu 2 menit 22 detik, menjadikannya sangat cocok untuk implementasi pada sistem deteksi warna atau perangkat dengan sumber daya terbatas. Faktor lain yang memengaruhi performa adalah pemilihan *optimizer*, *learning rate*, dan *batch size*. *Optimizer* Adam terbukti konsisten memberikan hasil terbaik karena kemampuannya menyesuaikan *learning rate* secara adaptif, sementara *learning rate* 0.001 menunjukkan kestabilan dan efektivitas lebih baik dibandingkan nilai yang lebih kecil. Meskipun *batch size* 32 dan 64 sama-sama menghasilkan performa yang baik tergantung arsitektur, *batch size* 32 terbukti optimal pada NASNet Mobile. Secara keseluruhan, NASNet Mobile direkomendasikan sebagai model terbaik untuk pengembangan lebih lanjut dalam sistem deteksi warna berbasis CNN karena mampu memberikan keseimbangan antara akurasi, efisiensi, dan kecepatan inferensi.

REFERENCES

[1] D. A. L. Sari, A. Mulyadi, A. Pratama, and R. Nalandari, "DETEKSI OBJEKBERWARNA REAL TIME BERDASARKAN VISUALISASI WEBCAM," *Zetroem*, vol. 02, no. 01, Mar. 2020.
[2] K. Angin, D. Teknik, and P. Citra, "Implementasi Metode K-Means Clustering untuk Mengklasterikasikan Kipas Angin dengan Teknik Pengolahan Citra," *J. Inform. Teknol. dan Sains*, vol. 7, no. 1, pp. 354–360, 2025.
[3] S. Saini, E. Febriani Dungga, and I. Sulistiani, "Evaluasi Pemeriksaan Tes Buta Warna Menggunakan Metode Ishihara Berbasis Google Form Menggunakan Buku Ishihara," *Indones. J. Pharm. Educ.*, vol. 2, no. 1, pp. 42–51, 2022, doi: 10.37311/ijpe.v2i1.15855.
[4] K. P. Lakshmi, A. Kalidindi, J. Chilukala, K. Nerella, W. Shaik, and D. Cherukuri, "A Color Guide for Color Blind People Using Image Processing and OpenCV," *Int. J. online Biomed. Eng.*, vol. 19, no. 9, pp. 30–46, 2023, doi: 10.3991/ijoe.v19i09.39177.
[5] J. Pardede, "Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection," *I.J. Intell. Syst. Appl.*, no. April, pp. 52–61, 2021, doi: 10.5815/ijisa.2021.02.04.
[6] N. A. Batubara and R. Maulana, *TUTORIAL OBJECT DETECTION PLATE NUMBER WITH CONVOLUTION NEURAL NETWORK (CNN)*. 2020.
[7] M. T. Rolly Maulana Awangga, S.T., M. K. Roni Andarsyah, S.T., and E. C. Putro, *Tutorial Object Detection People With Faster Region-Based Convolutional Neural Network (Faster R-CNN*. 2020.
[8] M. F. Naufal and S. F. Kusuma, "Analisis Perbandingan Algoritma Machine Learning dan Deep Learning untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 4, pp. 873–882, 2023, doi: 10.25126/jtiik.20241046823.
[9] R. L. Gaho, I. T. Ali, and E. Prakasa, "Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur Xception," *INOVTEK POLBENG*, vol. 9, no. 1, pp. 354–365, 2024.
[10] N. Hardi, U. Bina, S. Informatika, K. J. Pusat, and N. Mobile, "Komparasi Algoritma MobileNet Dan Nasnet Mobile Pada Klasifikasi Penyakit Daun Teh," *Rekayasa Perangkat Lunak*, vol. 3, no. 1, 2022.
[11] N. Saefulloh, J. Indra, and A. Ratna Juwita, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi

Kecacatan Pada Proses Welding di Perusahaan Manufacturing," *Technol. Sci.*, vol. 6, no. 1, pp. 387–394, 2024, doi: 10.47065/bits.v6i1.5321.

[12] S. Dhage, H. Chawan, A. Hoskote, A. Dabholkar, and V. Deshmukh, "Skin Cancer Classification Using Transfer Learning," *Lect. Notes Networks Syst.*, vol. 1080 LNNS, no. January, pp. 177–186, 2024, doi: 10.1007/978-3-031-67444-0_17.

[13] T. B. Sasongko, Haryoko, and A. Amrullah, "ANALISIS EFEK AUGMENTASI DATASET DAN FINE TUNE PADA ALGORITMAPRE-TRAINED CONVOLUTIONAL NEURAL NETWORK (CNN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 4, 2023.

[14] A. Y. Yulestiono, M. M. Subagio, M. S. Bhakti, and A. P. Sari, "Perbandingan Kinerja Metode Convolutional Neural Network (CNN) dan VGG-16 dalam Klasifikasi Rambu Lalu Lintas," *J. Mhs. Tek. Inform.*, vol. 3, no. 2, pp. 1–23, 2024.

[15] M. E. Purba, A. Z. Situmorang, G. L. B. Ginting, M. W. P. Lubis, and rans M. Sinaga, "Klasifikasi Sampah Organik dan Anorganik Menggunakan Algoritma CNN," *J. Sifo Mikroskil*, vol. 26, no. 1, 2025.

[16] W. Setiawan, *Deep Learning menggunakan Convolutional Neural Network: Teori dan Aplikasi*. 2020.

[17] R. Maulana, R. Dwi Zahra Putri, T. Ade Amelia, H. Syahputra, and F. Ramadhani, "Identifikasi Jenis Rempah-Rempah Indonesia Dengan Convolutional Neural Network (Cnn) Menggunakan Arsitektur Vgg16," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 6034–6039, 2024, doi: 10.36040/jati.v8i4.10138.

[18] S. K. Addagarla, "Real Time Multi-Scale Facial Mask Detection and Classification Using Deep Transfer Learning Techniques," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, 2020, doi: 10.30534/ijatcse/2020/33942020.

[19] A. T. Akbar, S. Saifullah, and H. Prapcoyo, "Klasifikasi Ekspresi Wajah Menggunakan Covolutional Neural NetworkKlasifikasi Ekspresi Wajah Menggunakan Covolutional Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 6, 2024.

[20] O. Adam, "Klasifikasi Hama Pada Daun Sawi Menggunakan Convolutional Neural Network (CNN) Dengan Algoritma Xcaption dan," *JEECOM (Journal Electr. Eng. Comput.*, vol. 6, no. 2, pp. 359–370, 2024, doi: 10.33650/jeecom.v4i2.