

Klasifikasi Sampah Organik dan Anorganik Menggunakan Algoritma CNN

Michael Emmanuel Purba¹, Angga Zefanya Situmorang², Geby Laurent Br Ginting³,
Muhammad Wahyu Pratama Lubis⁴, Frans Mikael Sinaga^{*5}

^{1,2,3,4} Universitas Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789

^{1,2,3,4} Fakultas Informatika, Teknik Informatika, Universitas Mikroskil, Medan

e-mail: 121111304@students.mikroskil.co.id, 211111287@students.mikroskil.co.id,

321111201@students.mikroskil.co.id, 4211110731@students.mikroskil.co.id,

5frans.sinaga@mikroskil.ac.id

Dikirim: 09-02-2025 | Diterima: 08-04-2025 | Diterbitkan: 30-04-2025

Abstrak

Penelitian ini mengembangkan sistem klasifikasi sampah menggunakan *algoritma Convolutional Neural Network (CNN)* untuk membedakan sampah organik dan anorganik. Arsitektur CNN yang digunakan terdiri dari input layer berukuran 150 x 150 x 3, tiga lapisan konvolusi dengan filter 3x3 yang jumlahnya bertambah secara bertahap (32, 64, dan 128), dilengkapi dengan fungsi aktivasi *ReLU* dan lapisan *MaxPooling 2x2*. Model juga menggunakan *dense layer* dengan 128 neuron, *dropout 0.5*, dan *output layer* dengan aktivasi *sigmoid*. Dalam proses pelatihan, model dikompilasi menggunakan *optimizer Adam* dengan *learning rate 0.001* dan *binary crossentropy* sebagai *loss function*. Hasil evaluasi menunjukkan bahwa model mencapai akurasi yang sangat baik sebesar 94.88% pada data uji, membuktikan efektivitas model dalam mengklasifikasikan sampah. Tingkat akurasi yang tinggi ini menunjukkan bahwa arsitektur CNN yang dirancang mampu mempelajari dan mengenali pola-pola penting dari citra sampah dengan baik. Penelitian ini memberikan kontribusi signifikan dalam pengembangan sistem pemilahan sampah otomatis yang dapat membantu meningkatkan efisiensi pengelolaan sampah. Model yang dikembangkan memiliki potensi besar untuk diimplementasikan dalam aplikasi praktis dan dapat diandalkan untuk sistem klasifikasi sampah yang efektif.

Kata kunci: *Convolutional Neural Network (CNN)*, Klasifikasi Sampah, *Deep Learning*, Pengolahan Citra Digital, Sistem Pemilahan Otomatis

Abstract

This research develops a waste classification system using the Convolutional Neural Network (CNN) algorithm to distinguish between organic and inorganic waste. The CNN architecture used consists of an input layer with dimensions of 150 x 150 x 3, three convolutional layers with 3x3 filters that gradually increase in number (32, 64, and 128), equipped with ReLU activation function and 2x2 MaxPooling layer. The model also uses a dense layer with 128 neurons, 0.5 dropout, and an output layer with sigmoid activation. In the training process, the model was compiled using Adam optimizer with a learning rate of 0.001 and binary crossentropy as the loss function. The evaluation results show that the model achieved excellent accuracy of 94.88% on test data, proving the model's effectiveness in classifying waste. This high level of accuracy indicates that the designed CNN architecture is capable of learning and recognizing important patterns from waste images well. This research makes a significant contribution to the development of automatic waste sorting systems that can help improve waste management efficiency. The developed model has great potential to be implemented in practical applications and can be relied upon for an effective waste classification system.

Keywords: *Convolutional Neural Network (CNN), Waste Classification, Deep Learning, Digital Image Processing, Automatic Sorting System*

1. PENDAHULUAN

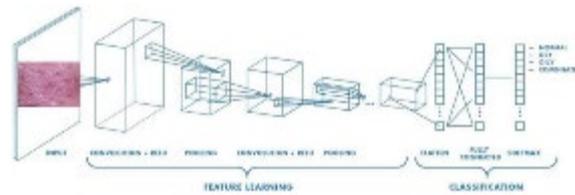
Sampah merupakan permasalahan yang dihadapi setiap negara. Setiap tahun jumlah dan jenis sampah meningkat seiring dengan bertambahnya jumlah penduduk. Di Indonesia, pada tahun 1990, 220 juta penduduk perkotaan menghasilkan kurang lebih 300.000 ton sampah per hari. Tahun 2000, 2,9 miliar orang yang tinggal di perkotaan menghasilkan 3 juta ton limbah sampah per hari. Pada tahun 2025 diperkirakan akan terus meningkat mencapai puluhan juta ton sampah per harinya [1]. Permasalahan sampah merupakan salah satu permasalahan lingkungan di masyarakat yang perlu ditangani dengan benar. Menurut *World Health Organization* (WHO), sampah adalah sesuatu yang sudah tidak dapat digunakan lagi, tidak dipakai, tidak disenangi, atau segala sesuatu yang dibuang berasal dari kegiatan manusia dan tidak terjadi sendirinya [2].

Sisa-sisa dari proses produksi, baik industri ataupun rumah tangga, yang sudah tidak terpakai tersebut biasa kita sebut sebagai sampah, di mana pada umumnya di Indonesia dikelompokkan menjadi 2 jenis, yaitu sampah anorganik dan sampah organik. Sampah yang berasal dari makhluk hidup merupakan sampah organik. Sampah organik mudah membusuk atau terurai hanya dalam waktu kurang dari 6 bulan. Sampah yang berasal dari sumber daya yang tidak dapat diperbarui adalah sampah anorganik. Sampah ini memiliki ciri sulit terurai sehingga butuh waktu hingga puluhan tahun agar dapat terurai dengan tanah. Di Indonesia, jumlah sampah yang dihasilkan dapat mencapai sekitar 11.330 ton/hari. Jika jumlah penduduk Indonesia di desa mencapai 128,5 juta jiwa dan di kota mencapai 126,3 juta jiwa, maka rata-rata setiap orang menghasilkan sampah sekitar 0,05 kg setiap harinya [3].

Pemilahan sampah di tingkat rumah tangga adalah salah satu kegiatan yang direkomendasikan oleh UU No. 18 Tahun 2008 tentang Pengelolaan Sampah. Pemilahan sampah itu sederhana, tapi efeknya cukup besar. Pemilahan sampah sangat membantu proses pengumpulan dan daur ulang terhadap sampah-sampah yang masih dapat dimanfaatkan kembali [4]. Salah satu aspek penting dalam pengelolaan sampah adalah pengelompokan sampah menjadi sampah organik dan anorganik. Klasifikasi yang efektif ini memungkinkan penerapan praktik daur ulang yang lebih efisien, meminimalkan dampak negatif terhadap lingkungan, dan memberikan dasar bagi pengembangan kebijakan pengelolaan limbah berkelanjutan. Dalam konteks ini, teknik kecerdasan buatan, khususnya *Convolutional Neural Network* (CNN), merupakan suatu jenis algoritma dalam *deep learning* yang dapat menerima gambar sebagai masukan (*input*) dan mengolahnya dengan tujuan untuk mengidentifikasi objek [5].

Convolutional Neural Network (CNN) merupakan perkembangan dari metode *Artificial Neural Network* (ANN) yang berfokus pada pemrosesan gambar, video, dan suara. CNN adalah salah satu metode *deep learning* yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada gambar digital. CNN telah banyak diterapkan di berbagai bidang, di antaranya untuk mendeteksi dan mengenali karakter pada plat nomor kendaraan. [6,7].

CNN (*Convolutional Neural Network*) merupakan salah satu jenis arsitektur jaringan saraf tiruan yang terdiri dari beberapa lapisan utama, seperti lapisan konvolusi untuk mengekstraksi fitur visual, lapisan *pooling* yang berfungsi mengurangi dimensi peta fitur, fungsi aktivasi untuk menambahkan nonlinieritas, serta lapisan *fully connected* yang berperan dalam proses klasifikasi atau regresi [8,9]. Sebagai bagian dari *Deep Neural Network*, CNN memiliki struktur mendalam yang dirancang untuk mengolah data berukuran besar [10]. Arsitektur *Convolutional Neural Network* secara umum pada gambar 1.

Gambar 1. *Arsitektur Convolutional Neural Network (CNN)*

Secara umum, arsitektur CNN terbagi menjadi empat tahap utama, yaitu input, ekstraksi fitur, klasifikasi, dan output. Proses ekstraksi fitur melibatkan lapisan-lapisan seperti konvolusi (*convolutional layer*), fungsi aktivasi (umumnya *ReLU*), dan *pooling*. Sementara itu, klasifikasi dilakukan melalui lapisan *fully connected* yang biasanya dikombinasikan dengan fungsi aktivasi seperti *Softmax*. CNN bekerja secara hierarkis, di mana keluaran dari satu lapisan menjadi masukan untuk lapisan berikutnya. Pada akhirnya, CNN menghasilkan keluaran berupa klasifikasi berdasarkan data input yang diberikan [11].

Fungsi aktivasi bukan dirancang untuk menonaktifkan, tetapi untuk mengaktifkan dan mengoptimalkan karakteristik neuron, memungkinkan fitur dipertahankan dan dipetakan melalui proses tersebut. Fungsi ini menentukan *output* neuron setelah menerima kumpulan input tertentu. Pada proses ini, jumlah berbobot dari nilai input jaringan diteruskan ke fungsi aktivasi untuk dilakukan konversi nonlinier.

Tahap ini berperan penting dalam menjaga informasi fitur yang relevan sekaligus menghilangkan redundansi data, menjadikannya elemen kunci dalam kemampuan jaringan saraf menyelesaikan masalah nonlinier [12]. Contoh beberapa fungsi aktivasi yang sering digunakan ditampilkan pada Gambar 2 di bawah.

Name	Function	Derivatives	Figure
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))^2$	
tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	
Leaky ReLU	$f(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	
Softmax	$f(x) = \frac{e^x}{\sum_j e^x}$	$f'(x) = \frac{e^x}{\sum_j e^x} - \frac{e^{2x}}{\sum_j e^{2x}}$	

Gambar 2. fungsi aktivasi yang sering digunakan

Penelitian ini akan menganalisis model CNN yang paling cocok untuk mengoptimalkan klasifikasi pemilahan sampah. Dengan algoritma ini, diharapkan proses pemisahan sampah secara otomatis dapat diterapkan dalam sistem. Tujuan dari penelitian ini adalah untuk mempercepat dan mempermudah proses pemilahan sampah dalam rangka mendukung tahapan daur ulang.

2. TINJAUAN PUSTAKA

2.1. Deteksi Objek

Deteksi objek adalah salah satu tugas utama dalam bidang visi computer yang bertujuan untuk mengidentifikasi dan memlokasi objek-objek tertentu dalam sebuah citra atau video. Tujuan utama dari deteksi objek adalah untuk mengenali dan menentukan posisi relatif dari objek-objek yang menarik dalam suatu gambar atau [13]. Deteksi objek memiliki banyak aplikasi praktis, termasuk pengawasan keamanan, kendaraan otonom, analisis video, pembelajaran, pengenalan wajah, deteksi kebakaran, identifikasi hama pada pertanian, pengenalan warna, dan banyak lagi.

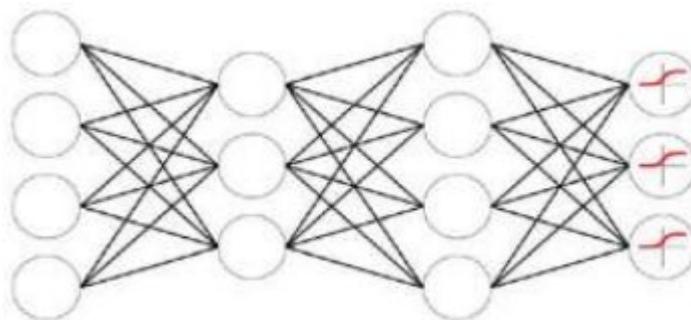
Proses deteksi objek melibatkan beberapa langkah, di antaranya:

1. Ekstraksi Fitur : Pada tahap ini, fitur-fitur relevan dari citra diekstraksi menggunakan teknik komputasi visual seperti *Convolutional Neural Networks* (CNN). Ekstraksi fitur ini dilakukan untuk memperoleh representasi numerik yang mewakili objek dalam citra.
2. Pembuatan Proposal : Setelah ekstraksi fitur, langkah selanjutnya adalah menghasilkan proposal atau kandidat lokasi objek dalam citra. Metode seperti *SelectiveSearch* atau *Region Proposal Networks* (RPN) digunakan untuk menghasilkan proposal berdasarkan ciri-ciri visual yang signifikan.
3. Pemilihan dan Pemrosesan Proposal : Pada tahap ini, proposal-proposal yang dihasilkan dievaluasi dan diproses untuk mengenali objek. Hal ini melibatkan penerapan model *machine learning* seperti *Support Vector Machines* (SVM) atau *Convolutional Neural Networks* (CNN) untuk mengklasifikasikan proposal berdasarkan kelas objek yang diinginkan.
4. Pemrosesan Posterior : Setelah klasifikasi, proposal-proposal yang terklasifikasi kemudian diproses lebih lanjut untuk meningkatkan akurasi dan keakuratan lokalisasi objek. Teknik seperti *Non-Maximum Suppression* (NMS) digunakan untuk menghapus proposal yang tumpang tindih dan mempertahankan proposal dengan skor tertinggi .

Lokalisasi dan Anotasi Objek : Pada langkah terakhir, objek-objek yang terdeteksi ditempatkan dalam kotak pembatas (*bounding box*) untuk menunjukkan lokasi dan ukuran objek. Informasi kelas dan atribut lainnya juga dapat ditambahkan sebagai anotasi pada kotak pembatas tersebut. [13].

2.2. Pembelajaran Mendalam

Pembelajaran mendalam (*Deep Learning*) adalah subbidang Pembelajaran Mesin berdasarkan jaringan saraf tiruan (JST) memberikan pengetahuan pada komputer melakukan suatu keputusan di dasari suatu kegiatan yang natural pada manusia. Komputer belajar mengkategorikan langsung dari gambar, teks, atau suara dalam pembelajaran mendalam. Komputer dilatih menggunakan sejumlah kumpulan data berlabel, dan nilai piksel gambar diubah menjadi representasi internal atau vektor fitur sehingga pengklasifikasi dapat mengenali atau mengkategorikan pola pada input [14]. Pada prinsipnya penggunaan algoritma *deep learning* (*Deep Neural Network*) ini adalah untuk mencari *feature engineering / feature extraction*. Keutamaan dari *deep learning* ini adalah merubah data dari *non-linear separable* menjadi *linearly separable* melalui serangkaian transformasi (*hidden layer*) [15].

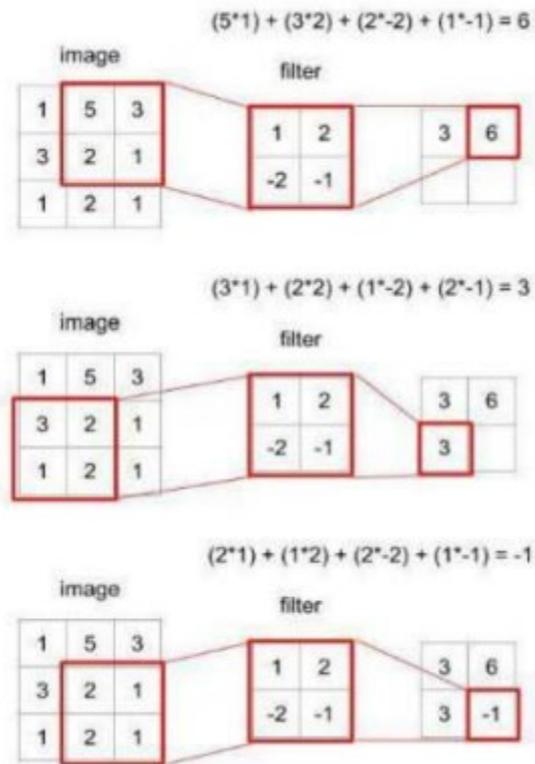


Gambar 3. *Multilayer Perceptron Deep Learning*

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN/*ConvNet*) 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. CNN dapat belajar langsung dari citra sehingga

mengurangi beban dari pemrograman [16]. CNN termasuk kedalam *neural network* bertipe *feed forward* (tidak berulang) [17]. Konsep utama CNN sendiri terdapat pada operasi konvolusi yang dimilikinya, dimana suatu citra akan diekstrasi setiap fiturnya agar terbentuk beberapa pola yang akan lebih mudah untuk diklasifikasi.

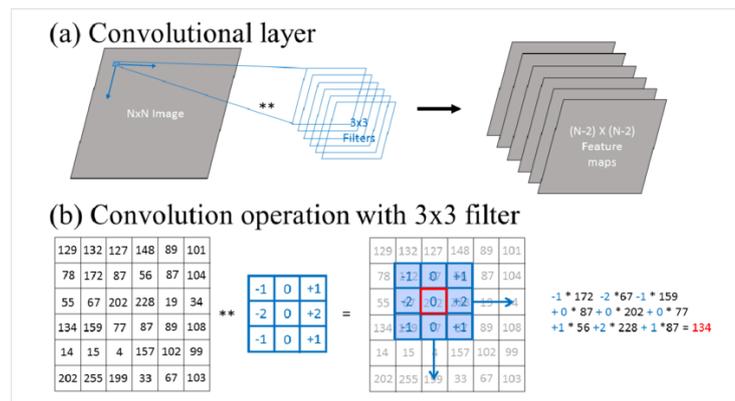


Gambar 4. Konsep *weight sharing* pada ekstraksi fitur konvolusi

Teknik ini dapat membuat fungsi pembelajaran gambar menjadi lebih efisien untuk diimplementasikan [18].

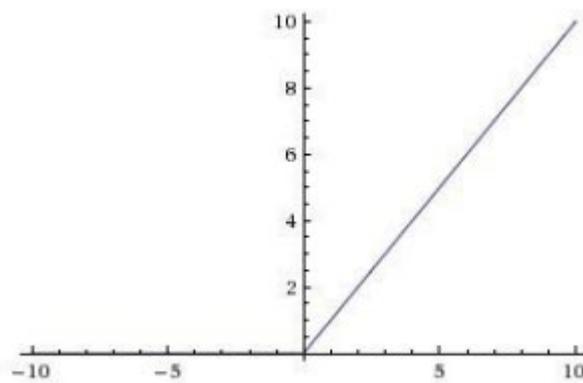
Secara garis besar CNN tidak jauh berbeda dengan *neural network*, neuron pada CNN memiliki *weight*, bias, dan *activation function*. Adapun lapisan penyusun dari sebuah CNN terdiri dari *Convolution Layer*, *Activation ReLU Layer*, *Pooling Layer*, dan *Fully Connected Layer*.

- a. *Convolution Layer* : *Convolution Layer* merupakan proses konvolusi yang menjadi bagian penting dalam arsitektur CNN, karena menjadi lapisan pertama yang akan memproses sebuah citra masukan pada sistem. Citra tersebut akan mengalami operasi konvolusi dan filtrasi untuk mendapatkan fitur ekstraksi dari citra masukan yang disebut *feature map* atau *activation map*. CNN menggunakan filter/aspek lokal paling informatif untuk mencari objek pada gambar. Filter tersebut akan terus digeser ke seluruh area/regional pada citra tersebut.

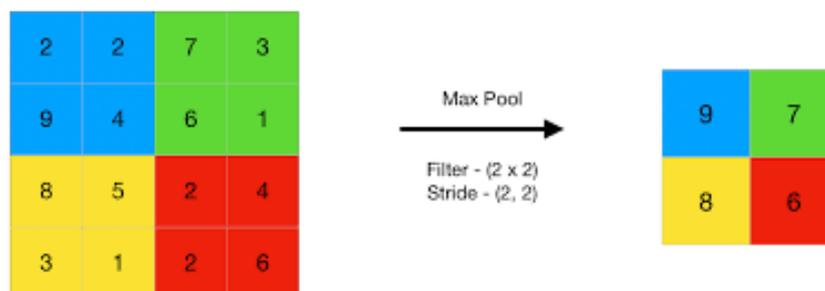


Gambar 5. Convolutional layer

- b. *Activation Rectified Linear Units (ReL-U)* : Aktivasi *ReL-U* berfungsi sebagai pengubah nilai piksel menjadi nol jika terdapat piksel pada citra tersebut memiliki nilai kurang dari nol dengan tujuan untuk meningkatkan kualitas dengan meminimalisir kesalahan.

Gambar 6. Aktivasi *ReL-U*

- c. *Pooling Layer* : Merupakan lapisan pada metode CNN yang menggunakan fungsi dimana *feature map* atau *activation map* sebagai masukan dan diteruskan untuk diolah dengan berbagai macam operasi statistik berdasarkan nilai piksel terdekat. Kelebihan dari *pooling layer* yaitu dapat secara progresif mengurangi ukuran *volume output* pada *feature map* yang dapat berguna untuk mengendalikan *overfitting*.

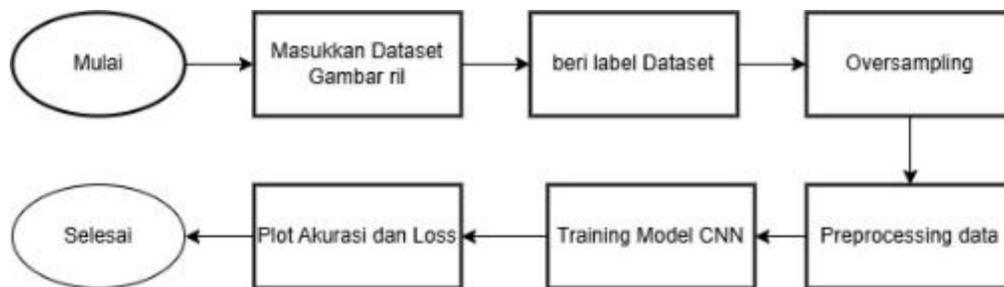
Gambar 7. *Pooling Layer*

- d. *Fully Connected Layer* : *Fully connected layer* merupakan kumpulan dari proses konvolusi. Neuron pada lapisan ini terhubung dengan semua neuron pada lapisan sebelumnya, dimana semua neuron di lapisan masukan perlu di ubah menjadi data satu dimensi. Lapisan ini mendapatkan input dari lapisan aktivasi sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu berdasarkan set data pelatihan.
- e. *Softmax Activation* : *Softmax Activation* merupakan bentuk lain dari *Logistic Regression* yang dapat digunakan untuk mengklasifikasi lebih dari dua kelas[3]. *Softmax* berguna untuk mengubah keluaran dari lapisan terakhir menjadi distribusi probabilitas dasarnya [19].

$$f_j(Z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (1)$$

3. METODE PENELITIAN

3.1 Tahapan Penelitian



Gambar 8. Tahapan Penelitian

1. Input: Memasukkan dataset gambar ril (*real image*) - ini adalah data mentah yang akan diproses oleh model yang akan dibangun
2. Pelabelan: Pemberian label pada dataset - mengidentifikasi dan menandai objek dalam gambar. Terdapat 2 label yaitu Organik dan Anorganik
3. *Oversampling*: Teknik untuk menyeimbangkan dataset yang tidak seimbang (*imbalanced*) - memastikan setiap kelas memiliki jumlah sampel yang setara. Dikarenakan data Organik memiliki jumlah yang sedikit ketimbang Anorganik sehingga dilakukan *oversampling* agar menambah akurasi dari model.
4. *Preprocessing data*: Persiapan data seperti normalisasi, *resize* gambar, dan augmentasi data bila diperlukan agar membantu model mengidentifikasi data dengan mudah
5. Training Model CNN: Proses pelatihan menggunakan *Convolutional Neural Network* - model belajar mengenali pola dari data
6. Plot Akurasi dan Loss: Visualisasi hasil performa model untuk melihat tingkat keberhasilan *training* dan validasi

3.2 Dataset dan Preprocessing

3.2.1 Dataset

Dataset yang digunakan merupakan dataset gambar. Dataset yang di pakai adalah data campuran yang diambil sendiri oleh peneliti dan data yang didapatkan dari situs *kaggle* lalu dipakai untuk melatih model. Dataset terdiri atas 2 jenis, yaitu :

- a. Dataset Organik : Dataset Organik berisikan Dataset yang organik cenderung terurai dengan lebih cepat. Jumlah Dataset Organik adalah 500 data.

- b. Dataset Anorganik: Dataset Anorganik berisikan dataset yang berbeda dengan organik karena dataset anorganik merupakan sampah yang membutuhkan waktu lebih lama untuk terurai. Jumlah dataset Anorganik adalah 489 data.

Tabel 1. Dataset yang telah di beri label dan di *oversampling*

No	Kelas	Train	Test	Total
1	Organik	400	100	500
2	Anorganik	400	100	500
Subtotal				1000

3.2.2 Preprocessing

Pada tahap *preprocessing* ini, tidak banyak menggunakan teknik *preprocessing* karena dataset yang diambil memiliki gambar yang bagus. Tetapi ada beberapa *preprocessing* gambar yang dilakukan, antara lain :

- Melakukan *image_size* dengan ukuran 200 x 200 pixel
- Mengatur *batch_size* menjadi 32.
- Pemisahan dataset menjadi dataset *train* dan dataset *test*
- Normalisasi nilai piksel gambar menjadi rentang [0,1]

3.3 Arsitektur Model

Metodologi penelitian ini menggunakan model *Convolutional Neural Network* (CNN) yang diimplementasikan menggunakan *library Keras*. Model CNN ini terdiri dari beberapa lapisan, seperti *data augmentation*, *MaxPooling2D*, *Conv2D*, *Dropout*, dan *Dense*. Arsitektur model secara lengkap didefinisikan, di mana fungsi loss yang digunakan adalah *binary_crossentropy* dan *optimizer* yang digunakan adalah *Adam* dengan *learning rate* 0,0001.

Proses pelatihan model dilakukan dengan menggunakan dataset pelatihan dan validasi. Dataset pelatihan dan validasi disiapkan dengan melakukan pemisahan data secara acak, di mana 80% data digunakan untuk pelatihan dan 20% data digunakan untuk validasi. Selama proses pelatihan, model dilatih selama beberapa *epoch* dengan *batch size* tertentu.

Penjelasan rinci mengenai setiap lapisan pada model CNN serta parameter-parameter yang digunakan diharapkan dapat membantu pembaca jurnal memahami pendekatan yang diambil dalam mengembangkan model untuk menyelesaikan permasalahan yang dibahas dalam penelitian ini.

```

cnn_model = tf.keras.models.Sequential([
    data_augmentation,
    tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid') # Satu neuron dengan sigmoid
])

cnn_model.compile(
    loss='binary_crossentropy', # binary_crossentropy untuk output biner
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
    metrics=['accuracy']
)

```

Gambar 9. Arsitektur Model

3.4 Pelatihan dan Validasi

3.4.1. Pelatihan

Proses pelatihan model CNN dilakukan dengan menggunakan fungsi *cnn_model.fit()*. Dataset pelatihan yang disiapkan sebelumnya, yang disimpan dalam variabel *train_data*, digunakan sebagai input untuk melatih model. Jumlah *epoch* pelatihan ditetapkan sebanyak 256 kali, di mana setiap *epoch* melewati seluruh dataset pelatihan 25 kali. Selain itu, dataset validasi yang disimpan dalam variabel *valid_data* digunakan untuk memantau performa model selama pelatihan.

Epoch 251/256	25/25	5s 90ms/step	- accuracy: 0.9614	- loss: 0.1174	- val_accuracy: 0.9500	- val_loss: 0.1856
Epoch 252/256	25/25	5s 99ms/step	- accuracy: 0.9544	- loss: 0.1271	- val_accuracy: 0.9400	- val_loss: 0.1606
Epoch 253/256	25/25	5s 99ms/step	- accuracy: 0.9448	- loss: 0.1399	- val_accuracy: 0.9300	- val_loss: 0.1685
Epoch 254/256	25/25	5s 105ms/step	- accuracy: 0.9339	- loss: 0.1791	- val_accuracy: 0.8950	- val_loss: 0.2981
Epoch 255/256	25/25	3s 111ms/step	- accuracy: 0.9445	- loss: 0.1278	- val_accuracy: 0.9100	- val_loss: 0.2289
Epoch 256/256	25/25	5s 97ms/step	- accuracy: 0.9489	- loss: 0.1485	- val_accuracy: 0.9400	- val_loss: 0.1729

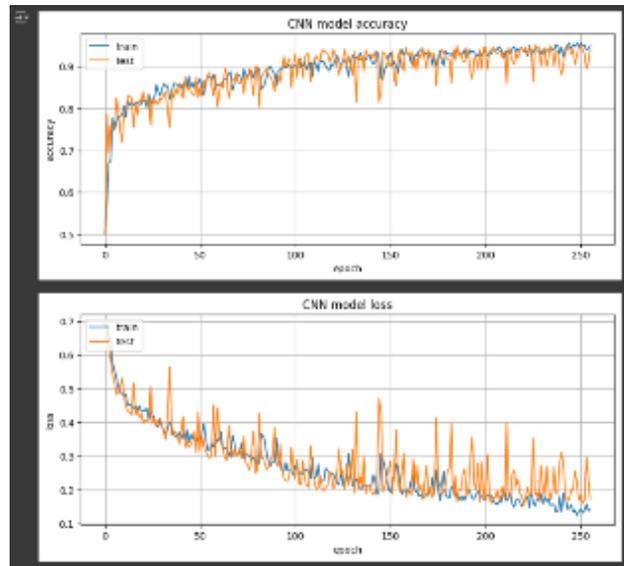
Gambar 10. Hasil Pelatihan Model

Berdasarkan hasil pelatihan model yang ditampilkan, dapat disimpulkan bahwa model menunjukkan peningkatan performa yang konsisten seiring dengan bertambahnya jumlah *epoch* pelatihan. Hal ini tercermin dari peningkatan akurasi dan penurunan nilai loss baik pada data pelatihan maupun data validasi. Hasil ini mengindikasikan bahwa model mampu belajar secara efektif dan tidak mengalami *overfitting* atau *underfitting* yang signifikan. Informasi kuantitatif yang disajikan dalam tabel dapat menjadi acuan yang berharga dalam mengevaluasi dan mengoptimalkan proses pelatihan model lebih lanjut.

3.4.2. Validasi

Grafik performa model CNN menunjukkan perkembangan yang positif selama proses pelatihan. Meskipun terdapat fluktuasi yang cukup besar pada akurasi dan nilai loss, secara keseluruhan model mampu mencapai akurasi yang cukup baik pada data pelatihan maupun data pengujian, serta mampu meminimalkan kesalahan prediksi. Hal ini mengindikasikan bahwa model dapat belajar secara efektif

tanpa mengalami *overfitting* yang parah, namun masih memerlukan pemantauan dan penyesuaian *hyperparameter* lebih lanjut untuk memperoleh performa yang optimal.



Gambar 11. Hasil Validasi pada akurasi dan nilai loss

3.5 Analisis Proses

1. Proses normalisasi data menggunakan *ImageDataGenerator* dengan parameter *rescale=1./255*, yang berarti nilai pikselnya dinormalkan dengan rentang [0, 1].
2. *Layer* pertama dengan
 - Ukuran kernel 3 x 3
 - Jumlah filter 32, yang berarti akan menghasilkan 32 *features maps*.
 - *Padding "same"*, yang berarti dimensi *output* akan sama dengan input,
 - Dimensi *output* yang ditentukan adalah 200 x 200 x 32
3. *Maxpooling* yang digunakan:
 - *Pool size 2 x 2*, yang akan mengurangi dimensi secara spasial.

Memakai rumus sebagai berikut :

$$Height = \frac{input\ dimension}{stride} \quad (2)$$

$$Weight = \frac{input\ dimension}{stride} \quad (3)$$

- *Output* dimensi dengan perhitungan:

$$Height = \frac{200}{2} = 100$$

dan,

$$Weight = \frac{200}{2} = 100$$

Sehingga, *output* yang didapat adalah 100 x 100 x 32

4. Konvolusi pada layer ke-2:
 - Ukuran kernel 3 x 3
 - Jumlah filter 64, dengan *Padding* yang tetap.

- Dimensi *output* menjadi 100 x 100 x 64
- 5. *Maxpooling* ke-2 dengan:
 - *Pool size* berukuran 2 x 2 dan dimensi spasial dikurangi ½.*Output* dimensi diperhitungkan menjadi:

$$\text{Height} = \frac{100}{2} = 50$$

dan,

$$\text{Weight} = \frac{100}{2} = 50$$

Sehingga, *output* yang didapat adalah 50 x 50 x 64.

- 6. Konvolusi pada *layer* ke-3, dengan ukuran filter 128, ukuran kernel 3 x 3, dan *Padding* yang masih sama, yaitu "same", di mana dimensi *outputnya* adalah 50 x 50 x 128.
- 7. *Maxpooling* ke-3 dengan *pool size* 2 x 2 dan dimensi spasial dikurangi ½ lagi, sehingga perhitungan *output* dimensi yaitu:

$$\text{Height} = \frac{50}{2} = 25$$

dan,

$$\text{Weight} = \frac{50}{2} = 25$$

Sehingga, *output* yang didapat adalah 25 x 25 x 128.

- 8. Lalu, neuron dinonaktifkan sebanyak 30% menggunakan fungsi *Dropout* tanpa mempengaruhi dimensi, dengan dimensi di tetap 25 x 25 x 128.
- 9. Masuk ke *layer Flatten*, untuk mengubah tensor multi dimensi yang dihasilkan dari proses konvolusi dan *maxpooling* sebelumnya menjadi vektor 1 dimensi agar dapat diproses oleh *layer fully connected (dense)*.

- *Output* dari *layer* sebelumnya berdimensi 25 x 25 x 128, maka perhitungan *flatten* yaitu:
Total Neuran = 25 x 25 x 128 = 80.000

Yang berarti ada 80.000 neuron yang menjadi *output* pada *layer* saat ini.

- 10. *Dense Layer* dengan jumlah neuron 128 dan menggunakan aktivasi *ReLU*, di mana rumus aktivasi *ReLU*:

$$z = w \cdot x + b \quad (4)$$

di mana, **w** adalah bobot, **x** adalah input dari *layer* sebelumnya, dan **b** adalah bias, dengan *output* dimensi pada *layer* ini adalah 128

- 11. *Layer fully connected* kedua untuk menyaring dan menyederhanakan dari proses sebelumnya dengan hanya 64 neuron dan menggunakan aktivasi yang sama, yaitu *ReLU*. Sehingga perhitungan dimensi yaitu:

$$z = w \cdot x + b$$

atau,

$$\alpha = \max(0, z) \quad (5)$$

Di mana, α adalah hasil setelah fungsi aktivasi *ReLU* diterapkan dan *output* dimensi adalah 64.

- 12. *Layer Dense* terakhir arsitektur CNN pada penelitian ini menggunakan 1 neuron dengan fungsi aktivasi *sigmoid*, karena model nantinya hanya akan menghasilkan 1 nilai yg memrepresentasikan probabilitas dengan rentang 0 dan 1.

Fungsi aktivasi *sigmoid*:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

Sehingga, *output* dimensi yang dihasilkan adalah 1.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Tampilan Aplikasi

Berikut ini adalah hasil tampilan aplikasi sesuai dari perancangan yang telah dibuat:

1. Tampilan Utama

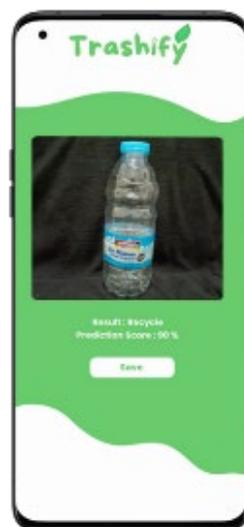
Pada tampilan utama aplikasi, terdapat tombol yang memungkinkan pengguna untuk mengunggah gambar dari galeri ke dalam aplikasi. Setelah gambar berhasil dimasukkan, pengguna dapat menekan tombol "Analisis" untuk memproses gambar tersebut. Hasil analisis akan menampilkan jenis sampah, apakah termasuk kategori organik atau anorganik, berdasarkan gambar yang telah diunggah.



Gambar 12. Tampilan Utama

2. Tampilan Hasil

Pada tampilan hasil, pengguna akan melihat gambar yang sebelumnya telah diunggah. Gambar tersebut kemudian dianalisis oleh model yang telah dikembangkan untuk mengidentifikasi jenis sampah, apakah organik atau anorganik. Selain itu, hasil analisis akan menampilkan persentase akurasi yang mencerminkan tingkat keyakinan model terhadap klasifikasi yang diberikan.

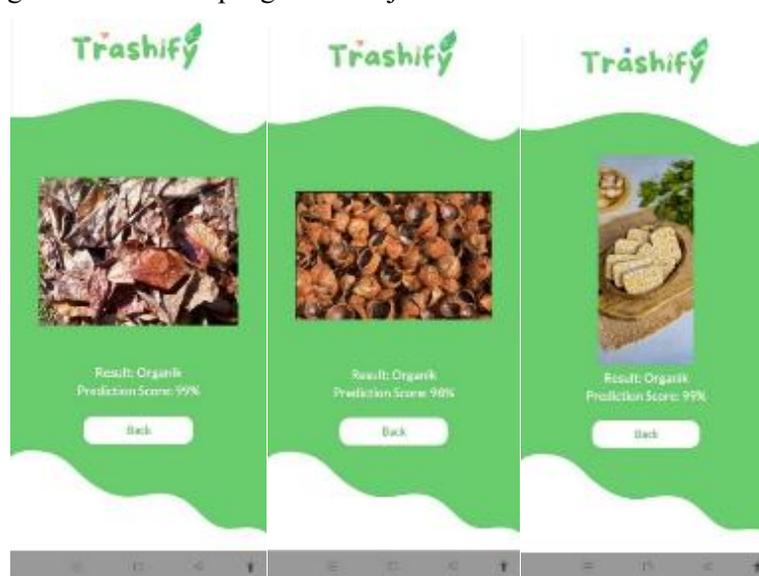


Gambar 13. Tampilan hasil

4.2. Hasil Pengujian Model pada Aplikasi

1. Organik

Hasil pengujian model pada aplikasi untuk klasifikasi gambar jenis organik menunjukkan kinerja yang sangat baik. Model ini berhasil mengklasifikasikan gambar dengan tingkat akurasi yang sangat tinggi, yaitu mencapai rata-rata 97,6%. Angka ini menggambarkan kemampuan model dalam membedakan objek organik dari kategori lainnya dengan sangat presisi, memberikan indikasi bahwa model telah dilatih dengan baik dan mampu mengenali pola-pola visual yang khas pada gambar organik. Pengujian ini menunjukkan potensi model untuk diterapkan dalam pengklasifikasian gambar organik dalam skala yang lebih luas, dengan harapan dapat meningkatkan efisiensi dan akurasi dalam berbagai aplikasi terkait pengolahan citra dan pengenalan objek.



Gambar 14. Hasil pengujian model pada aplikasi untuk sampah organik

2. Anorganik

Hasil pengujian model pada aplikasi untuk klasifikasi gambar jenis anorganik menunjukkan performa yang sangat unggul. Model mampu mengklasifikasikan gambar dengan tingkat akurasi rata-

rata sebesar 98,4%. Hal ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengenali dan membedakan pola visual yang khas pada gambar anorganik. Tingginya tingkat akurasi ini mengindikasikan bahwa model telah berhasil menangkap fitur-fitur penting dari data pelatihan, sehingga dapat diandalkan untuk mendeteksi kategori anorganik secara konsisten. Dengan performa tersebut, model ini memiliki potensi besar untuk diterapkan dalam berbagai aplikasi berbasis pengolahan citra, terutama dalam konteks yang membutuhkan klasifikasi objek anorganik dengan tingkat presisi yang tinggi.



Gambar 15. Hasil pengujian model pada aplikasi untuk sampah anorganik

4.3. Analisis Kinerja Model

Hasil evaluasi model yang telah dilampirkan, dapat disimpulkan bahwa arsitektur model CNN yang digunakan terdiri dari *input layer* berukuran $150 \times 150 \times 3$, tiga lapisan konvolusi dengan jumlah filter yang meningkat, dilanjutkan dengan lapisan *MaxPooling*, *Flatten*, *dense layer* dengan *dropout*, dan *output layer* dengan aktivasi *sigmoid*. Model CNN tersebut dikompilasi menggunakan *optimizer Adam*, *loss function binary crossentropy*, dan akurasi sebagai metrik evaluasi.

Overall Training Accuracy: 94.88%
Overall Validation Accuracy: 94.00%

Gambar 16. Hasil Keseluruhan *Training* dan *Validation*

Berdasarkan hasil evaluasi, model mencapai akurasi hingga 94.88% pada data uji, menunjukkan bahwa model CNN yang dilatih mampu mengklasifikasikan objek menjadi dua kelas (organik dan anorganik) dengan tingkat akurasi yang cukup baik. Selain akurasi, hasil evaluasi juga menampilkan nilai *loss* dan akurasi per *epoch*, yang dapat digunakan untuk menganalisis tren performa model selama proses pelatihan. Proses perhitungan dapat dilihat pada rumus 7 [20].

$$Accuracy = \frac{True\ Positive}{Sum\ of\ Data} \quad (7)$$

Secara keseluruhan, pendekatan menggunakan arsitektur CNN yang dirancang dengan baik dan dilatih dengan parameter yang sesuai telah menghasilkan model klasifikasi dengan performa yang memuaskan pada dataset yang digunakan dalam penelitian ini.

5. KESIMPULAN

Hasil pengujian akhir terhadap model CNN yang dilatih menunjukkan performa yang sangat baik dengan mencapai akurasi sebesar 94,88% pada data uji, menandakan bahwa model ini sangat efektif dalam mengklasifikasikan sampah organik dan anorganik. Tingkat akurasi yang tinggi ini membuktikan bahwa arsitektur CNN yang dirancang mampu mempelajari dan mengenali pola-pola penting dari citra sampah dengan baik. Hasil ini juga menunjukkan bahwa model yang dikembangkan memiliki potensi yang besar untuk diimplementasikan dalam sistem pemilahan sampah otomatis, yang dapat membantu meningkatkan efisiensi pengelolaan sampah. Meskipun model telah menunjukkan kinerja yang memuaskan, masih terdapat peluang untuk pengembangan lebih lanjut melalui penambahan dataset dan penyesuaian parameter untuk meningkatkan generalisasi model. Dengan demikian, penelitian ini telah berhasil mengembangkan sebuah sistem klasifikasi sampah yang dapat diandalkan dan siap untuk diimplementasikan.

6. SARAN

Berdasarkan hasil penelitian ini, terdapat beberapa rekomendasi untuk pengembangan lebih lanjut. Pertama, perlu dilakukan perluasan dataset dengan menambahkan berbagai gambar sampah dalam kondisi yang beragam, seperti variasi pencahayaan, sudut pengambilan gambar, dan latar belakang, agar model memiliki kemampuan generalisasi yang lebih baik. Kedua, pengujian model pada data dunia nyata, seperti gambar sampah yang diambil langsung dari lingkungan sekitar, sangat disarankan untuk mengevaluasi kinerjanya dalam situasi yang lebih kompleks dan beragam.

Selanjutnya, penelitian di masa depan dapat difokuskan pada pengoptimalan *hyperparameter*, termasuk jumlah *filter*, ukuran *batch*, *learning rate*, dan jumlah *epoch*, untuk lebih meningkatkan akurasi model. Integrasi model dengan teknologi berbasis *Internet of Things* (IoT), seperti sensor kamera pada tempat sampah pintar, juga dapat menjadi langkah strategis untuk menciptakan sistem pemilahan sampah otomatis yang lebih efisien.

Di samping itu, pengembangan model dapat diarahkan untuk mencakup klasifikasi sampah berdasarkan jenis material yang lebih spesifik, seperti plastik, logam, kaca, atau kertas, sehingga model lebih relevan dalam pengelolaan sampah. Eksplorasi arsitektur *deep learning* yang lebih kompleks, seperti *ResNet* atau *EfficientNet*, juga dapat dipertimbangkan untuk menghasilkan performa yang lebih optimal dalam mengenali pola data dengan tingkat kompleksitas yang tinggi.

Terakhir, model perlu dievaluasi secara rutin untuk memastikan kinerjanya tetap optimal, terutama jika diterapkan pada skala yang lebih besar atau dalam lingkungan yang berbeda. Dengan menerapkan rekomendasi ini, diharapkan model yang dikembangkan dapat lebih efektif, memiliki kegunaan praktis yang lebih luas, dan memberikan kontribusi yang signifikan dalam mendukung pengelolaan sampah yang berkelanjutan.

DAFTAR PUSTAKA

- [1] K. A. Priana and A. A. I. N. E. Karyawati, "Sistem Pendeteksi Sampah Secara Realtime Menggunakan Metode YOLO," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 2, no. 1, pp. 31–36, 2023.
- [2] K. C. A. Unwaru, Sudarti, and Yushardi, "Analisis Pemanfaatan Sampah Organik Dan Anorganik Terhadap Teknologi PLTSa Di Beberapa Kota Besar Di Indonesia," *Jurnal Sains Dan Teknologi*, vol. 5, no. 2, pp. 255–263, 2022.
- [3] F. P. Fantara, D. Syauqy, and G. E. Setyawan, "Implementasi Sistem Klasifikasi Sampah Organik dan Anorganik dengan Metode Jaringan Saraf Tiruan Backpropagation," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, vol. 2, no. 11, pp. 5577–5586, 2018.

- [4] R. Khoirunnisa Apriyani, N. Rustanti, D. Puspa Rahayu, and N. Dewi Utami Hamid, “Sosialisasi Pengenalan Dan Pemilahan Jenis Sampah Organik Dan Anorganik Di Panti Asuhan Anak Shaleh,” *Jurnal Pengabdian Kepada Masyarakat*, vol. 1, no. 2, pp. 43–60, 2023, doi: 10.59820/pengmas.v1i2.46.
- [5] G. A. Bahagia and M. Akbar, “KLASIFIKASI SAMPAH ORGANIK DAN ANORGANIK MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” vol. 8, no. 5, pp. 10349–10355, 2024.
- [6] N. H. Harani, C. Prianto, and M. Hasanah, “Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python,” *Jurnal Teknik Informatika*, vol. 11, no. 3, pp. 47–53, 2019, [Online]. Available: <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/658>
- [7] Edwin Febrywinata, “Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab,” *Merkurius: Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 2, no. 4, pp. 185–193, 2024, doi: 10.61132/mercurius.v2i4.162.
- [8] B. Lintang Wibowo, O. Yoga Caesarizky, M. Ilham Hakim, and M. Munsarif, “Application of convolutional neural network (CNN) algorithm to classify vehicle category Penerapan algoritma convolutional neural network (cnn) untuk mengklasifikasi jenis kendaraan,” *Jurnal Komputer dan Teknologi Informasi*, vol. 2, no. 2, pp. 94–99, 2024, doi: 10.26714/jkti.v.
- [9] A. Hardirega and I. Jaelani, “IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) KLASIFIKASI MOTIF BATIK MENGGUNAKAN EFFICIENTNET-B1,” vol. 8, no. 5, pp. 10023–10028, 2024.
- [10] M. M. Sebatubun and C. Haryawan, “Implementasi Algoritma Convolutional Neural Network untuk Klasifikasi Jenis Keris,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 3, pp. 595–602, 2024, doi: 10.25126/jtiik.937260.
- [11] M. Dandi Darajat, Y. A. Sari, and R. C. Wihandika, “Convolutional Neural Network untuk Klasifikasi Citra Makanan Khas Indonesia,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, pp. 4764–4769, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] Z. Xu, S. Qian, X. Ran, and J. Zhou, “Application of Deep Convolution Neural Network in Crack Identification,” *Applied Artificial Intelligence*, vol. 36, no. 1, pp. 229–232, 2022, doi: 10.1080/08839514.2021.2014188.
- [13] D. Nafis Alfarizi, R. Agung Pangestu, D. Aditya, M. Adi Setiawan, and P. Rosyani, “Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis,” *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [14] P. Musa, W. K. Anam, S. B. Musa, W. Aryunani, R. Senjaya, and P. Sularsih, “Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network,” *Rekayasa*, vol. 16, no. 1, pp. 65–73, 2023, doi: 10.21107/rekayasa.v16i1.16974.
- [15] M. D. Payana, D. Ria, Y. Tb, Z. Musliyana, and M. B. Wibawa, “Deteksi Masker Wajah Menggunakan Metode Convolutional Neural Network (CNN) Meningkatkan Nilai Akurasi Melalui Arsitektur Layer Konvolusi,” *Journal of Informatics and Computer Science*, vol. 8, no. 1, pp. 30–35, 2022.
- [16] S. Ilahiyah and A. Nilogiri, “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network _ Ilahiyah _ JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia),” *JUSTINDO(Jurnal Sistem & Teknologi Informasi Indonesia)*, vol. 3, no. 2, pp. 49–56, 2018.
- [17] P. A. Nugroho, I. Fenriana, and R. Arijanto, “Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia,” *Algor*, vol. 2, no. 1, pp. 12–21, 2020.

- [18] M. Fasounaki, E. B. Yüce, S. Öncül, and G. Ince, “CNN-based Text-independent Automatic Speaker Identification Using Short Utterances,” *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, vol. 01, pp. 413–418, 2021, doi: 10.1109/UBMK52708.2021.9559031.
- [19] M. A. Hanin, R. Patmasari, and R. Y. Nur, “Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (Cnn) Skin Disease Classification System Using Convolutional Neural Network (Cnn),” *e-Proceeding of Engineering*, vol. 8, no. 1, pp. 273–281, 2021.
- [20] G. Thiodorus, A. Prasetia, L. A. Ardhani, and N. Yudistira, “Klasifikasi citra makanan/non makanan menggunakan metode Transfer Learning dengan model Residual Network,” *Teknologi*, vol. 11, no. 2, pp. 74–83, 2021, doi: 10.26594/teknologi.v11i2.2402.

