JURNAL ILMIAH SISTEM INFORMASI (JUSI)
VOL 4, No. 3, November 2025, pp. 802 - 817
p-ISSN : 2803-1507 (media cetak) | e-ISSN : 2803-1531 (media online)
*Link Publikasi: https://ejurnal.provisi.ac.id/index.php/JUISI/issue/view/151*

# Identification of Road Damage Using the Convolutional Neural Network (CNN) Method

**Ahmad Syafiq Maulana Zuhri*[1], Nur Nafiiyah[2], Agus Setia Budi[3]**
[1,2,3] Fakultas Sains dan Teknologi, Universitas Islam Lamongan
Jalan Veteran No.53A, 62211, Jetis, Lamongan, Kabupaten Lamongan, Jawa Timur, Indonesia
e-mail: syafiqmaulanazuhri@gmail.com*[1], mynaff@unisla.ac.id[2], geniusbudi@unisla.ac.id[3]

**ARTICLE INFO**

*Article History:*

*Received: 22-08-2025*
*Revised: 02-09-2025*
*Accepted: 13-10-2025*
*Published: 30-10-2025*

**ABSTRACT**

*Road damage is a common issue that directly affects the safety and comfort of road users, as well as transportation efficiency. Traditionally, road inspections are carried out manually, requiring considerable time, cost, and resources. This study aimed to develop an automatic classification system to detect road damage from surface images, utilizing the Convolutional Neural Network (CNN) method. The dataset used in this research consists of 400 images with a balanced distribution, comprising 200 images of the Cracks category (damaged roads) and 200 images of the non-Cracks category (undamaged roads), obtained from an open dataset on the Mendeley Data platform. The CNN architecture was designed with four convolutional layers equipped with ReLU activation functions, pooling layers, dropout layers to reduce overfitting, and a fully connected layer for the final classification stage. The training process was conducted using TensorFlow and Keras on Google Colab, with the dataset split into 80% for training and 20% for validation. Evaluation results showed that the model achieved excellent classification performance. Based on the average across training scenarios (epochs 30, 40, and 50), the CNN model achieved an overall accuracy of 97.50%, an average precision of 96.77%, recall of 99.37%, and F1-score of 97.72%. With consistent performance and a low error rate, this CNN model has strong potential as an automated tool for image-based road damage identification, enabling faster inspections, reducing manual workload, and assisting agencies in making timely, well-informed infrastructure maintenance decisions.*

**Keywords:** *CNN, Image Classification, Road Damage, Deep Learning, Model Evaluation.*

**Abstrak**

Kerusakan jalan merupakan permasalahan umum yang berdampak langsung terhadap keselamatan dan kenyamanan pengguna jalan, serta terhadap efisiensi transportasi. Selama ini, proses inspeksi jalan masih dilakukan secara manual, yang memerlukan waktu, biaya, dan sumber daya yang tidak sedikit. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis guna mendeteksi kerusakan jalan berdasarkan citra permukaan, dengan memanfaatkan metode Convolutional Neural Network (CNN). Dataset yang digunakan dalam penelitian ini terdiri dari 400 gambar dengan distribusi seimbang, yaitu 200 gambar kategori Cracks (jalan retak) dan 200 gambar kategori non-Cracks (jalan tidak retak), yang diambil dari sumber dataset

DOI: https://doi.org/10.51903/jgn2sd35

p-ISSN : 2803-1507 e-ISSN : 2803-1531

terbuka di platform Mendeley Data. Arsitektur CNN dirancang secara khusus dengan empat lapisan konvolusi yang dilengkapi fungsi aktivasi ReLU, pooling layer, dropout layer untuk mengurangi overfitting, serta fully connected layer pada tahap akhir klasifikasi. Proses pelatihan dilakukan menggunakan TensorFlow dan Keras di platform Google Colab, dengan pembagian data sebesar 80% untuk data pelatihan dan 20% untuk data validasi. Hasil evaluasi menunjukkan bahwa model memiliki performa klasifikasi yang sangat baik. Berdasarkan rata-rata dari seluruh skenario pelatihan (epoch 30, 40, dan 50), model CNN yang dikembangkan mampu mencapai akurasi keseluruhan sebesar 97,50%, presisi rata-rata 96,77%, recall rata-rata 99,37%, dan F1-score rata-rata 97,72%. Dengan kinerja yang konsisten dan tingkat kesalahan yang rendah, model CNN ini memiliki potensi besar untuk diterapkan sebagai alat bantu dalam proses identifikasi kerusakan jalan berbasis citra secara otomatis dan efisien, sehingga dapat mempercepat inspeksi, mengurangi beban kerja manual, dan membantu instansi terkait dalam pengambilan keputusan pemeliharaan infrastruktur jalan secara tepat waktu.

**Kata Kunci**: *CNN, Klasifikasi Citra, Kerusakan Jalan, Deep Learning, Evaluasi Model.*

## 1. PENDAHULUAN

Jalan merupakan salah satu kebutuhan krusial dalam aktivitas masyarakat karena berperan penting dalam mempermudah akses transportasi dari tempat asal ke tempat tujuan. Selain itu, jalan juga berfungsi sebagai infrastruktur transportasi darat yang menunjang mobilitas manusia dan distribusi barang. Kondisi jalan yang baik sangat dibutuhkan agar kegiatan masyarakat dapat berjalan lancar. Sebaliknya, jalan yang tidak layak dapat menimbulkan berbagai permasalahan, salah satunya meningkatnya risiko kecelakaan lalu lintas. Jalan yang menghubungkan satu wilayah dengan wilayah lainnya disebut jalan raya. Keberadaan jalan raya memiliki peran penting dalam mendukung aktivitas sosial dan ekonomi masyarakat, terutama dalam sektor transportasi [1], [2]

kecelakaan di Indonesia disebabkan oleh berbagai faktor termasuk kerusakan jalan. Untuk mengurangi faktor-faktor tersebut survei manual dilakukan secara berkala untuk mendeteksi kerusakan jalan, namun survei manual membuat perbaikan jalan terhambat dan tidak merata, studi ini melihat sistem deteksi kerusakan jalan untuk membantu mengklasifikasikan dan menilai kondisi jalan, serta bahan evaluasi dan penilaian untuk digunakan oleh instansi terkait [3]. Sistem ini diharapkan dapat meningkatkan efisiensi proses pemantauan infrastruktur melalui analisis citra secara otomatis.

Identifikasi retakan jalan yang efektif dan tepat waktu sangat penting untuk memperbaiki dan membatasi degradasi lebih lanjut. Hingga saat ini, sebagian besar metode deteksi retakan mengikuti pendekatan inspeksi manual yang bertentangan dengan deteksi berbasis gambar otomatis, membuat keseluruhan prosedur mahal dan memakan waktu [4]. Pendekatan manual ini juga mengandalkan keahlian individu yang dapat menimbulkan ketidakakuratan atau kelalaian dalam identifikasi kerusakan jalan. Penggunaan metode berbasis *image processing* memungkinkan hasil deteksi yang lebih objektif dan dapat direplikasi dengan mudah.

Penerapan hasil penelitian secara langsung pada skenario nyata dinilai sulit bagi pengelola jalan. Untuk mengatasi keterbatasan tersebut, penelitian ini menyusun kumpulan data kerusakan jalan berskala besar yang baru, lalu melatih dan mengevaluasi model deteksi kerusakan menggunakan metode *Convolutional Neural Network* (CNN) yang lebih canggih [5]. Model tersebut diuji dengan berbagai kondisi pencahayaan dan permukaan jalan untuk menilai tingkat keandalannya.

Dengan adanya kemajuan dalam bidang pengolahan citra dan algoritma deep learning, seperti *Convolutional Neural Network* (CNN), sekarang memungkinkan untuk melakukan deteksi kerusakan jalan secara otomatis menggunakan citra yang diambil dari kamera atau drone. CNN memiliki kemampuan untuk mengekstrak fitur secara otomatis dari gambar, yang memungkinkan deteksi kerusakan permukaan jalan dengan tingkat akurasi yang tinggi. Pendekatan ini dapat mengurangi ketergantungan pada inspeksi manual yang membutuhkan tenaga dan waktu besar. Selain itu, sistem berbasis *deep learning* mampu beradaptasi terhadap variasi kondisi lingkungan dan tipe kerusakan jalan yang berbeda.

## 2. TINJAUAN PUSTAKA

Penelitian mengenai deteksi dan klasifikasi kerusakan jalan berbasis pengolahan citra dan *deep learning* telah dilakukan secara luas dengan berbagai pendekatan model. Menurut [6], penggunaan citra permukaan jalan berformat RGB dengan tahap *preprocessing* berupa konversi ke *grayscale*, *resize*, dan *edge detection* mampu

p-ISSN : 2803-1507 e-ISSN : 2803-1531

meningkatkan kemampuan CNN dalam mengenali pola retakan jalan. Penelitian [7] membangun dataset gabungan dari sumber publik dan internal untuk klasifikasi trotoar retak dan tidak retak dengan akurasi di atas 96%. Sementara itu, penelitian [8] menerapkan empat model *preprocessing* berbeda dan menemukan bahwa konversi citra ke *grayscale* menghasilkan akurasi terbaik hingga 99% pada data validasi.

Penelitian [9] mengimplementasikan arsitektur CNN berbasis *transfer learning* VGG16 dengan tiga skenario *preprocessing* menggunakan kombinasi CLAHE, *Gamma Correction*, dan *Green Channel*. Skenario kedua, yang menggabungkan CLAHE dan *Gamma Correction*, memberikan akurasi 82,98%. Penelitian [10] mengembangkan sistem deteksi lubang jalan secara *real-time* berbasis YOLOv11 dengan *precision* 75,1%, *recall* 76,6%, dan *F1-score* 75,8%. Implementasi sistem berbasis web memberikan kemudahan akses bagi pengguna dalam mendeteksi dan memperkirakan luas kerusakan jalan secara otomatis.

Penelitian [11] menyoroti pentingnya pembangunan dataset yang sesuai dengan konteks lokal untuk mendukung model deteksi kerusakan jalan berbasis CNN. Studi [12] menjelaskan pentingnya deteksi dini terhadap kerusakan infrastruktur jalan untuk menunjang efektivitas pemeliharaan. Selain itu, penelitian [13] mengkaji penerapan *Deep Learning* dalam berbagai bidang seperti pengenalan objek visual dan deteksi citra, menunjukkan efektivitas metode ini untuk data visual kompleks. Penelitian [14] menegaskan bahwa CNN unggul karena mampu melakukan ekstraksi fitur secara otomatis tanpa metode manual. Selanjutnya, penelitian [15] menyebutkan bahwa CNN merupakan pengembangan dari MLP yang menggunakan prinsip konvolusi untuk mengekstraksi fitur penting dari data visual. Penelitian [16] menunjukkan bahwa struktur CNN terinspirasi dari sistem saraf manusia, memungkinkan pengenalan pola visual yang efisien.

Penelitian [17] menguraikan bahwa *Convolution Layer* berfungsi mengekstraksi fitur dari citra input untuk mengenali pola visual seperti tekstur atau bentuk. Penelitian [18] menekankan pentingnya *Pooling Layer* dalam mengurangi dimensi data untuk mempercepat komputasi. Studi [19] memperjelas bahwa *Pooling Layer* juga membuat model lebih toleran terhadap distorsi atau pergeseran kecil pada citra. Penelitian [20] membahas penerapan *Batch Normalization* yang mempercepat pelatihan model dan meningkatkan stabilitas distribusi input antar lapisan. Studi [21] menjelaskan bahwa *Dropout* digunakan untuk mengurangi *overfitting* dengan cara menonaktifkan neuron secara acak selama pelatihan, membuat model lebih kuat terhadap variasi data. Penelitian [22] menjelaskan bahwa *Fully Connected Layer* digunakan untuk menggabungkan semua fitur hasil ekstraksi sebelum klasifikasi akhir dilakukan. Penelitian [23] menguraikan konsep *optimizer* sebagai algoritma penyesuaian bobot model untuk meminimalkan kesalahan prediksi. Sementara itu, penelitian [24] menunjukkan bahwa penggunaan *Optimizer Adam* dapat mempercepat pelatihan dan meningkatkan akurasi model CNN dalam mendeteksi pola visual.

Penelitian [25] mengembangkan model CNN berbasis *transfer learning* untuk klasifikasi permukaan jalan dengan hasil akurasi tinggi. Studi [26] mengadaptasi pendekatan CNN dan *dropout optimization* untuk meningkatkan efisiensi komputasi pada perangkat dengan sumber daya terbatas. Penelitian [27] mengeksplorasi penerapan arsitektur CNN dengan teknik *feature extraction* dan *fine-tuning*, yang terbukti meningkatkan akurasi klasifikasi citra jalan secara signifikan. Berdasarkan seluruh penelitian terdahulu tersebut, dapat disimpulkan bahwa penggunaan CNN dan metode *deep learning* memiliki potensi besar dalam mendukung sistem pemantauan infrastruktur jalan yang efisien, akurat, dan mudah diimplementasikan.

## 3. METODOLOGI PENELITIAN

### 3.1. *Alur Penelitian*

Penelitian ini bertujuan mengklasifikasikan citra jalan menjadi dua kategori, yaitu *Cracks* (retak) dan *non-Cracks* (tidak retak), menggunakan model *Convolutional Neural Network* (CNN) yang dibangun dan dilatih dari awal tanpa transfer learning maupun arsitektur pra-latih seperti ResNet, VGG, atau MobileNet. Arsitektur CNN dirancang dengan empat *convolution layer* agar mampu mengenali fitur dari yang sederhana hingga pola retakan kompleks. Jumlah filter ditingkatkan secara bertahap untuk memperkaya representasi fitur, sementara *learning rate* kecil (0.00001) dipilih guna mencegah overfitting pada dataset yang relatif terbatas. Dengan pendekatan ini, model dikembangkan secara khusus sesuai kebutuhan klasifikasi dua kelas dalam penelitian. Bagan alur penelitian identifikasi tingkat kerusakan jalan menggunakan CNN secara keseluruhan adalah sebagaimana yang ditampilkan pada Gambar 1.

p-ISSN : 2803-1507 e-ISSN : 2803-1531




**Gambar 1.** Alur Penelitian

### 3.1.1. *Pengumpulan Dataset*

*Dataset* berasal dari Mendeley Data (Endri et al., 2020) dengan total 400 gambar: 200 Cracks dan 200 non-Cracks (*dataset* seimbang). Setiap gambar dikumpulkan dari permukaan jalan dengan kondisi pencahayaan dan tekstur yang bervariasi untuk memastikan model mampu mengenali berbagai pola retakan. Data tersebut sudah melalui proses penyaringan agar tidak terdapat gambar yang buram atau tidak relevan. Kualitas gambar dijaga agar resolusi cukup tinggi sehingga pola retakan dapat terdeteksi dengan baik.

### 3.1.2. *Pembagian Dataset*

Data dibagi menjadi 80% latih (320 gambar) dan 20% validasi (80 gambar) untuk melatih model dan mengukur generalisasi. Pembagian ini dilakukan secara acak agar tidak terjadi bias dalam proses pelatihan. Proporsi tersebut dipilih karena memberikan keseimbangan antara kebutuhan data untuk pembelajaran dan data untuk evaluasi. Dengan pembagian ini, model dapat belajar dari variasi data yang cukup luas sekaligus diuji pada data yang belum pernah dilihat sebelumnya.

### 3.1.3. *Pelatihan Model CNN*

CNN dibangun dari nol dengan 4 *convolution layer* (*filter* 8, 16, 32, 64; *kernel* 3x3), aktivasi *ReLU*, *max pooling*, *dropout* (0.5), *flatten*, dan *dense layer* dengan *softmax*. *Optimizer* Adam dengan *learning rate* 0.00001 digunakan untuk mempercepat konvergensi model. Setiap *epoch* pelatihan dilakukan dengan ukuran *batch* tertentu agar model mampu menyesuaikan bobot secara efisien. Arsitektur ini dirancang agar jaringan dapat mengekstraksi fitur penting secara bertahap dari lapisan awal hingga lapisan akhir.

### 3.1.4. *Evaluasi Performa Model*

Evaluasi menggunakan *accuracy*, *precision*, *recall*, *f1-score*, serta *confusion matrix* untuk analisis hasil prediksi. Metrik-metrik ini dipilih karena mampu memberikan gambaran menyeluruh mengenai performa model pada data seimbang. Setiap nilai metrik dihitung berdasarkan hasil prediksi terhadap data validasi. Analisis ini membantu dalam menilai seberapa baik model mengenali kelas *Cracks* dan *non-Cracks* secara konsisten.

### 3.1.5. *Implementasi*

Model disimpan dalam format .h5 dan diintegrasikan ke aplikasi berbasis web (*Streamlit*) agar pengguna dapat mengunggah gambar jalan dan memperoleh hasil klasifikasi otomatis (*Cracks* / *non-Cracks*). Aplikasi ini dirancang dengan antarmuka sederhana sehingga mudah digunakan oleh pengguna umum maupun peneliti. Proses klasifikasi dilakukan secara *real-time* dengan menampilkan hasil dan tingkat keyakinan model. Integrasi ini memungkinkan penerapan model dalam konteks pemantauan kondisi jalan secara praktis.

## 3.2. *Rancangan Data*

Pada tahap ini, dilakukan perencanaan dan pengelolaan data yang digunakan dalam penelitian klasifikasi tingkat kerusakan jalan. Rancangan data mencakup proses pengumpulan *dataset* dan *dataset splitting*, agar sesuai dengan kebutuhan model pembelajaran mesin berbasis *Convolutional Neural Network* (CNN). Pemilihan *dataset* yang tepat, serta proses pengolahan awal yang sistematis, sangat penting untuk memastikan bahwa model CNN dapat mempelajari pola-pola visual yang relevan, seperti retakan atau permukaan jalan yang mulus, secara optimal. Tahap rancangan data ini juga membantu memastikan ketersediaan data yang representatif sehingga hasil pelatihan model dapat mencerminkan kondisi nyata di lapangan.

### 3.2.1. *Pengumpulan Dataset*

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Dataset yang digunakan diambil dari penelitian [15] yang tersedia secara terbuka di platform *Mendeley Data* dengan alamat https://data.mendeley.com/datasets/xnzhj3x8v4/2, dataset ini mempunyai dua kelas yakni *Cracks* dan *non-Cracks,* masing-masing kelas berisi 200 gambar, sehingga total dataset adalah sebanyak 400 gambar, oleh karena itu dalam dataset ini tidak perlu untuk dilakukan penyeimbangan data karena dataset sudah memiliki distribusi yang seimbang antara kedua kelas, yaitu kelas *Cracks* dan *non-Cracks*.

Dalam proses pengumpulan dan analisis data, setiap kelas dalam *dataset* memiliki karakteristik visual yang berbeda yang penting untuk dipahami sebelum pelatihan model dilakukan. Salah satu kelas yang menjadi fokus utama adalah kelas *Cracks*, yang merepresentasikan kondisi permukaan jalan dengan kerusakan berupa retakan. Informasi mengenai bentuk, pola, dan tekstur retakan diperlukan agar model dapat mengenali fitur-fitur utama yang membedakan citra rusak dari yang tidak rusak. Contoh citra dari kelas tersebut ditunjukkan pada Gambar 2, yang menampilkan data dari kelas *Cracks*.



**Gambar 2.** Dataset Kelas Cracks

Gambar 2, menunjukkan kelas Cracks dalam dataset yang terdiri dari gambar permukaan jalan yang memiliki kerusakan berupa retakan. Ciri-ciri utamanya meliputi garis-garis retak memanjang, bercabang, atau berpola jaring pada aspal, dengan tekstur permukaan yang tidak mulus. Retakan bisa tipis maupun tebal, dan terkadang membentuk pola kompleks seperti alligator crack. Model CNN perlu mengenali pola-pola ini agar mampu membedakan retakan nyata dari elemen lain seperti bayangan atau garis marka jalan.

Selain kelas *Cracks*, *dataset* juga mencakup kelas lain yang merepresentasikan kondisi jalan tanpa kerusakan. Kelas ini memiliki peran penting sebagai pembanding agar model dapat membedakan antara permukaan jalan yang mengalami retakan dan yang masih utuh. Pemahaman terhadap karakteristik visual kelas ini membantu meningkatkan akurasi proses pelatihan model *CNN*. Contoh citra dari kelas tersebut dapat dilihat pada *Gambar 3*, yang menampilkan data dari kelas *non-Cracks*.



**Gambar 3.** Dataset Kelas Non-Cracks

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Gambar 3 menunjukkan kelas *non-Cracks* dalam *dataset*, yakni berisi gambar permukaan jalan yang tidak mengalami kerusakan atau retakan. Ciri-cirinya mencakup permukaan aspal yang mulus, rata, dan tanpa garis pecah atau pola retakan. Meski begitu, gambar di kelas ini tetap dapat memperlihatkan variasi tekstur alami, perubahan warna karena cahaya, bayangan, atau keberadaan marka jalan, yang tidak termasuk dalam kategori kerusakan. Model CNN harus mampu membedakan elemen-elemen visual ini agar tidak salah mengklasifikasikannya sebagai retakan.

### 3.2.2. *Pembagian Dataset*

Dalam penelitian [28], pelatihan model *EfficientNet-B0* pada skenario pembagian 80% data *training* dan 20% data *validasi* mampu memberikan performa yang lebih unggul untuk jenis model tersebut. Pada skenario ini, model dengan penerapan *fine-tuning* menghasilkan akurasi tertinggi sebesar 0.9846 dengan nilai *loss* terendah yaitu 0.0904. Hasil ini menunjukkan bahwa pemilihan proporsi data yang tepat berpengaruh terhadap kemampuan model dalam mengenali pola visual secara efektif. Proses pembagian data yang seimbang juga membantu mengontrol kualitas pelatihan agar model tidak hanya menghafal pola dari data latih. Rincian pembagian jumlah data pada masing-masing kelas ditunjukkan pada Tabel 1.

**Tabel 1.** Pembagian Dataset

| No | Kelas | Train | Val | Total |
|---|---|---|---|---|
| 1 | *Cracks* | 160 | 40 | 200 |
| 2 | *Non-Cracks* | 160 | 40 | 200 |
| Total | | 320 | 80 | 400 |

Pada Tabel 1, dalam penelitian ini *dataset* juga dibagi menjadi dua bagian utama, yaitu data latih (*training data*) dan data validasi (*validation data*), dengan rasio pembagian sebesar 80:20. Dari pembagian tersebut, sebanyak 320 gambar digunakan sebagai data latih yang berfungsi untuk melatih model dalam mengenali pola visual dari masing-masing kelas. Sementara itu, 80 gambar lainnya digunakan sebagai data validasi, yang bertujuan untuk mengevaluasi performa model selama proses pelatihan berlangsung. Pembagian data ini dilakukan agar model dapat belajar secara optimal dari sebagian besar data, sekaligus diuji kemampuannya pada data yang belum pernah dilihat sebelumnya, sehingga dapat diketahui seberapa baik model melakukan generalisasi terhadap citra baru.

## 3.3. *Tahapan Pelaksanaan Penelitian*

Tahapan pelaksanaan penelitian ini mencakup serangkaian proses sistematis yang bertujuan untuk membangun, menguji, dan mengimplementasikan model klasifikasi citra permukaan jalan berbasis *deep learning*, khususnya menggunakan arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan secara mandiri. Setiap tahapan dirancang untuk mendukung keberhasilan pengembangan sistem, mulai dari tahap pemodelan awal hingga pengujian akhir dan pembuatan aplikasi berbasis web. Metode ini digunakan agar setiap tahap penelitian dapat saling berhubungan dan menghasilkan keluaran yang sesuai dengan tujuan penelitian. Selain itu, pendekatan sistematis membantu dalam mengurangi potensi kesalahan selama proses pengembangan model.

Tahapan ini meliputi pembangunan model CNN dengan arsitektur yang dirancang sendiri, pengaturan *hyperparameter* seperti jumlah *epoch* dan *batch size* untuk memperoleh performa optimal, evaluasi performa model berdasarkan metrik klasifikasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, penyimpanan model (*export model*) yang telah terlatih untuk kebutuhan implementasi, serta pengembangan aplikasi berbasis web sebagai bentuk realisasi sistem klasifikasi citra permukaan jalan. Setiap tahapan dilakukan secara berurutan agar hasil dari satu tahap dapat menjadi masukan bagi tahap berikutnya. Pengaturan *hyperparameter* dilakukan melalui serangkaian percobaan untuk mendapatkan kombinasi parameter terbaik. Proses penyimpanan model dilakukan setelah sistem mencapai kinerja optimal untuk memastikan hasil pelatihan dapat dimanfaatkan pada implementasi berikutnya.

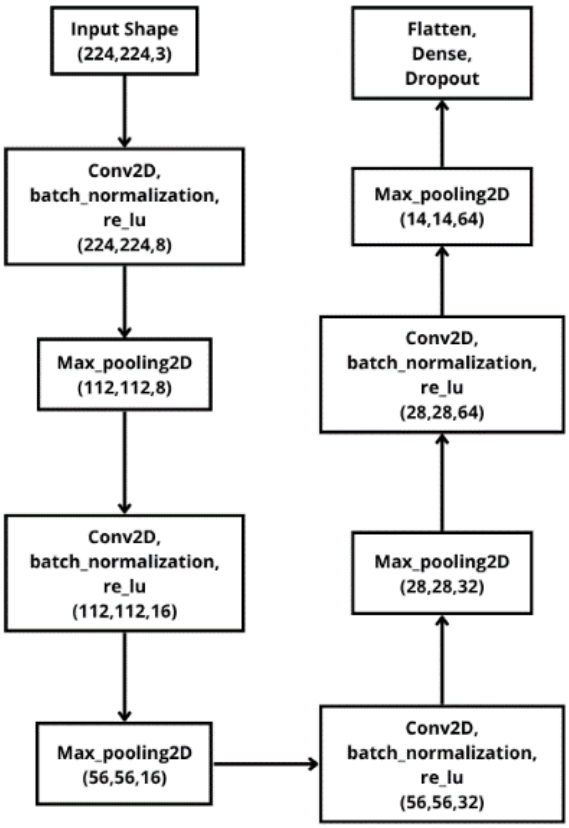
### 3.3.1. *Training Model*

Pada tahap ini dilakukan proses pelatihan model *Convolutional Neural Network* (CNN) menggunakan data citra permukaan jalan yang telah diproses sebelumnya. Tujuan dari proses pelatihan (*training*) ini adalah untuk membuat model mampu mengenali dan mempelajari pola-pola visual yang terdapat pada gambar jalan, sehingga dapat membedakan antara dua kelas, yaitu *Cracks* (retak) dan *non-Cracks* (tidak retak). Proses ini menjadi tahap awal dalam pengembangan sistem klasifikasi citra karena kualitas pelatihan akan sangat

p-ISSN : 2803-1507 e-ISSN : 2803-1531

memengaruhi kemampuan model dalam mendeteksi pola visual pada data baru. Selain itu, pelatihan dilakukan dengan memperhatikan parameter yang relevan seperti ukuran *batch*, jumlah *epoch*, dan tingkat *learning rate* agar model dapat beradaptasi dengan karakteristik citra permukaan jalan.

Sebelum pelatihan, dataset citra terlebih dahulu dibagi menjadi dua bagian, yaitu data latih (*training data*) dan data validasi (*validation data*). Data latih digunakan untuk mengajarkan model terhadap berbagai variasi citra yang mewakili kedua kelas, sedangkan data validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan. Selain itu, selama pelatihan, diterapkan beberapa teknik tambahan seperti optimasi parameter model melalui algoritma *backpropagation* dan *optimizer* seperti Adam. Proses ini dilakukan secara iteratif dalam beberapa *epoch* hingga model mencapai tingkat akurasi dan *loss* yang optimal. Hasil dari tahap pelatihan ini kemudian divisualisasikan dalam bentuk bagan arsitektur CNN seperti yang ditunjukkan pada Gambar 4.




**Gambar 4.** Struktur Model CNN

Bagan struktur model CNN yang digunakan dalam penelitian ini pada Gambar 4 yang dibuat berdasarkan penelitian dari [14], menunjukan bahwa Input gambar memiliki ukuran (224, 224, 3), yaitu gambar dengan resolusi 224x224 piksel dan 3 saluran warna (RGB). Ini lebih spesifik dan digunakan untuk dataset tertentu, menggunakan flatten untuk mengubah data dari dimensi multi-dimensi (output konvolusi) menjadi vektor satu dimensi sebelum diteruskan ke layer dense, adanya dropout yang mengurangi overfitting dan jumlah filter lebih banyak hingga 64. Model ini dirancang untuk meningkatkan efisiensi dalam proses ekstraksi fitur dari data citra. Setiap komponen dalam arsitektur tersebut memiliki peran penting dalam memastikan bahwa informasi spasial pada gambar tetap terjaga selama proses konvolusi berlangsung.

Dalam penelitian ini, pelatihan model Convolutional Neural Network (CNN) menggunakan Google Colab dan memakai Tensorflow/Keras dalam bahasa python, untuk pelatihan model arsitektur CNN penelitian ini menggunakan optimizer Adam (Adaptive Moment Estimation) dengan nilai learning rate sebesar 0.00001. Pemilihan Adam karena kemampuannya menyesuaikan pembaruan bobot secara adaptif, sehingga lebih efisien dan stabil. Learning rate yang rendah dipilih untuk menghindari pembelajaran yang terlalu cepat dan mencegah overfitting, sehingga model dapat belajar secara bertahap dan optimal dalam mengenali pola pada

p-ISSN : 2803-1507 e-ISSN : 2803-1531

data citra jalan, Model yang digunakan dalam penelitian ini memiliki arsitektur yang terdiri dari empat layer konvolusi. Setiap layer menggunakan kernel berukuran 3x3 dan jumlah filter yang meningkat secara bertahap, yaitu 8, 16, 32, dan 64 filter.

### 3.3.2. *Evaluasi*

Evaluasi performa model CNN tidak hanya bergantung pada satu matriks, tetapi seringkali melibatkan beberapa matriks yang saling melengkapi untuk memberikan gambaran yang lebih lengkap tentang kemampuan model, penelitian ini mencakup empat aspek, yaitu, *true Positive* (TP) menunjukkan prediksi positif yang tepat, *true Negative* (TN) menunjukkan prediksi negatif yang tepat, *false Positive* (FP) menunjukkan prediksi positif yang salah, *false Negative* (FN) menunjukkan prediksi negatif yang salah. Setiap matriks tersebut memiliki fungsi dan peran yang berbeda dalam menggambarkan kualitas hasil prediksi model. Penggunaan kombinasi dari beberapa matriks ini dapat membantu peneliti memahami kelebihan dan kelemahan model secara lebih komprehensif. Penjelasan lebih detail mengenai beberapa matriks evaluasi utama sebagai berikut:

Nilai akurasi menunjukkan seberapa sering model menghasilkan prediksi yang benar terhadap seluruh data yang diuji. Meskipun sering digunakan, akurasi bisa menyesatkan ketika distribusi kelas tidak seimbang. Dalam kondisi seperti itu, model dapat tampak akurat padahal gagal mengenali sebagian kelas tertentu. Akurasi merupakan rasio prediksi benar (positif dan negatif) terhadap keseluruhan data [2]. Persamaan *accuracy* pada Rumus (1).

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

Metrik *precision* menggambarkan sejauh mana prediksi positif yang diberikan model benar-benar relevan. Nilai *precision* yang tinggi menunjukkan bahwa model jarang memberikan prediksi positif yang salah. Metrik ini sangat penting dalam kasus di mana kesalahan positif memberikan dampak yang signifikan. *Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif [2]. Persamaan *precision* ditunjukkan pada Rumus (2).

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

Metrik *recall* menilai sejauh mana model mampu mendeteksi seluruh data positif tanpa terlewat. Nilai *recall* yang tinggi menandakan bahwa model berhasil mengenali sebagian besar data positif. Metrik ini sangat relevan ketika kehilangan satu data positif dianggap sebagai kesalahan yang besar. *Recall* mengukur seberapa baik model dalam mengidentifikasi seluruh sampel yang sebenarnya positif. Persamaan *recall* ditunjukkan pada Rumus (3).

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

Metrik *F1-Score* berfungsi untuk menilai performa model secara menyeluruh dengan mempertimbangkan dua aspek utama, yaitu ketepatan dan kelengkapan. Nilai *F1-Score* yang tinggi menunjukkan bahwa model mampu menjaga keseimbangan antara kemampuan mendeteksi dan keakuratan prediksi. Metrik ini sangat berguna ketika terdapat ketidakseimbangan kelas, karena mempertimbangkan baik *false positives* maupun *false negatives*. *F1-Score* merupakan rata-rata harmonis antara *precision* dan *recall*, sehingga memberikan keseimbangan antara kedua matriks tersebut. Persamaan *F1-Score* ditunjukkan pada Rumus (4).

$$F1\ Score = 2x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (2)$$

Dalam evaluasi model CNN, pemilihan metrik harus disesuaikan dengan tujuan tugas dan karakteristik data. Untuk klasifikasi gambar, kombinasi metrik seperti *accuracy*, *precision*, *recall*, dan *F1-Score* sering digunakan. Setiap jenis metrik memiliki kekuatan dan kelemahan yang berbeda sesuai dengan konteks data yang digunakan. Analisis yang tepat terhadap hasil evaluasi dapat membantu meningkatkan kinerja model secara signifikan.

### 3.3.3. *Implementasi*

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Selanjutnya, dilakukan tahap implementasi metode *Convolutional Neural Network (CNN)*, yang dimulai setelah diperoleh model klasifikasi terbaik dari proses pelatihan sebelumnya. Model yang telah mencapai performa optimal kemudian disimpan dalam format *.h5*. Format ini memungkinkan model untuk dimuat kembali di kemudian hari tanpa perlu dilakukan pelatihan ulang, sehingga efisien dari segi waktu dan sumber daya. Penyimpanan model dalam format tersebut juga memastikan bahwa seluruh parameter dan bobot yang telah dipelajari selama proses pelatihan tetap terjaga dengan baik.

Setelah model disimpan, langkah berikutnya dalam penelitian ini adalah menerapkannya ke dalam sebuah sistem identifikasi tingkat kerusakan jalan berbasis *web*. Sistem ini dirancang agar pengguna dapat dengan mudah mengunggah gambar permukaan jalan, dan model akan secara otomatis mengidentifikasi apakah gambar tersebut termasuk dalam kategori retak (*cracks*) atau tidak retak (*non-cracks*). Implementasi berbasis *web* dipilih karena memberikan kemudahan akses bagi pengguna tanpa memerlukan instalasi perangkat lunak tambahan. Antarmuka sistem dibuat sederhana agar proses unggah gambar dan penentuan hasil klasifikasi dapat dilakukan dengan cepat dan intuitif.

## 4. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari proses implementasi metode *Convolutional Neural Network (CNN)* dalam mengidentifikasi tingkat kerusakan jalan berdasarkan citra permukaan aspal. Penelitian dilakukan dengan menggunakan *dataset* berjumlah 400 gambar yang terdiri dari dua kelas, yaitu *Cracks* (jalan rusak) dan *non-Cracks* (jalan tidak rusak). Model CNN ini dipilih karena memiliki kemampuan tinggi dalam mengenali pola visual yang kompleks pada citra permukaan jalan. Setiap citra dalam *dataset* telah melalui tahap *preprocessing* untuk menyesuaikan ukuran dan kontras sehingga memudahkan model dalam proses pembelajaran.

Model CNN yang digunakan dibangun secara mandiri dengan empat lapisan konvolusi, kemudian dilatih menggunakan bahasa pemrograman Python dan *framework TensorFlow/Keras* pada platform *Google Colab*. Proses pelatihan dilakukan dalam beberapa skenario jumlah *epoch*, yaitu 30, 40, dan 50, guna mengetahui pengaruh jumlah *epoch* terhadap performa model. Setiap skenario pelatihan menghasilkan tingkat akurasi dan *loss* yang berbeda, sehingga pengujian dilakukan secara menyeluruh untuk melihat kestabilan performa. Selain itu, parameter pelatihan seperti ukuran *batch* dan fungsi aktivasi juga disesuaikan untuk memastikan model dapat belajar secara optimal dari data yang tersedia.

Hasil yang diperoleh meliputi matriks evaluasi seperti *accuracy*, *loss*, *precision*, *recall*, dan *F1-score*, serta visualisasi berupa grafik pelatihan dan *confusion matrix*. Seluruh hasil ini digunakan untuk menganalisis sejauh mana model mampu mengenali pola kerusakan pada citra jalan, serta untuk menentukan konfigurasi pelatihan terbaik yang dapat digunakan pada tahap implementasi aplikasi berbasis web. Grafik pelatihan menunjukkan tren peningkatan akurasi dan penurunan *loss* seiring bertambahnya jumlah *epoch*. Hasil evaluasi juga memperlihatkan bahwa model mampu melakukan klasifikasi dengan tingkat ketepatan yang konsisten pada data validasi.

### 4.1. *Implementasi Model CNN*

Penelitian ini berfokus pada penerapan *Convolutional Neural Network (CNN)* untuk mengklasifikasikan citra permukaan jalan ke dalam dua kategori, yaitu *Cracks* (retak) dan *non-Cracks* (tidak retak), menggunakan *dataset* berjumlah 400 gambar yang terbagi merata. *Dataset* tersebut dipisahkan menjadi 80% data latih dan 20% data validasi agar model dapat dilatih sekaligus dievaluasi pada data baru. Model CNN dibangun dari awal dengan empat *convolution layer* berfilter 8, 16, 32, dan 64 dengan *kernel* 3x3, dilengkapi aktivasi *ReLU*, *max pooling* untuk mereduksi dimensi, *dropout* untuk mencegah *overfitting*, serta *flatten* dan *dense layer* beraktivasi *sigmoid* untuk menghasilkan prediksi dua kelas. Arsitektur tersebut dirancang agar mampu menangkap fitur tekstur dan pola retakan pada citra aspal secara lebih mendalam melalui kombinasi lapisan konvolusi yang berurutan.

Proses pelatihan menggunakan *optimizer Adam* dengan *learning rate* 0.00001, sehingga pembaruan bobot berjalan stabil dan model dapat belajar bertahap tanpa *overfitting*. Setelah pelatihan selesai, model disimpan dalam format *.h5* agar dapat digunakan kembali tanpa perlu dilatih ulang. Model ini kemudian diimplementasikan pada aplikasi berbasis web menggunakan *framework Streamlit*, yang memungkinkan pengguna mengunggah citra jalan dan memperoleh hasil klasifikasi secara otomatis dengan cepat dan praktis. Aplikasi tersebut juga dilengkapi antarmuka yang ramah pengguna sehingga memudahkan proses pengujian langsung di lapangan menggunakan perangkat komputer maupun ponsel.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

### 4.2. *Proses Training Model Epoch 30, 40, 50*

Dalam penelitian ini, proses *training* model *Convolutional Neural Network (CNN)* dilakukan dengan tiga skenario jumlah *epoch*, yaitu 30, 40, dan 50 *epoch*. Tujuan variasi ini adalah untuk melihat pengaruh jumlah iterasi pelatihan terhadap performa model. Model *CNN* dipilih karena memiliki kemampuan yang baik dalam mengenali pola spasial pada data citra. Setiap skenario *epoch* dilakukan dengan parameter dan arsitektur model yang sama untuk menjaga konsistensi hasil eksperimen. Hasil dari proses pelatihan pada skenario *epoch* 30 ditampilkan untuk memberikan gambaran awal mengenai performa model, seperti ditunjukkan pada Tabel 2.

**Tabel 2.** Proses Training Epoch 30

| Train Ke- | Val_accuracy | Val_Loss | Train_accuracy | Train_loss |
|---|---|---|---|---|
| 1 | 96,25% | 0,157 | 99,06% | 0,055 |
| 2 | 96,25% | 0,149 | 99,37% | 0,060 |
| 3 | 95,00% | 0,136 | 99,06% | 0,066 |
| 4 | 93,75% | 0,129 | 98,75% | 0,070 |
| 5 | 96,25% | 0,139 | 99,37% | 0,063 |
| ***Avg*** | **95,50%** | **0,142** | **99,12%** | **0,063** |

Seperti yang ditunjukan pada Tabel 2, proses *training* ke-5 dengan *epoch* 30 menunjukkan nilai yang paling baik di antara hasil *training* lainnya pada *epoch* 30. Kemudian ditunjukkan bahwa pelatihan model menghasilkan rata-rata *train_accuracy* sebesar 99,12% dan rata-rata *val_accuracy* sebesar 95,50%. Nilai akurasi yang tinggi menunjukkan bahwa model mampu belajar dengan baik dari data latih yang diberikan. Meskipun demikian, nilai *validation accuracy* yang sedikit lebih rendah dibandingkan *training accuracy* menandakan adanya kemungkinan *gap* performa antara data latih dan data validasi. Hal ini mengindikasikan bahwa model mulai mencapai titik optimal pada *epoch* ke-30.

Skenario berikutnya dilakukan dengan jumlah *epoch* sebanyak 40 untuk mengetahui apakah peningkatan jumlah iterasi mampu memperbaiki performa model. Dalam tahap ini, struktur model, fungsi aktivasi, serta algoritma optimisasi tetap dipertahankan sama dengan skenario sebelumnya agar hasil yang diperoleh dapat dibandingkan secara objektif. Evaluasi dilakukan dengan menghitung nilai *accuracy* dan *loss* pada data pelatihan serta validasi. Hasil dari proses pelatihan pada skenario *epoch* 40 dapat dilihat seperti ditunjukkan pada Tabel 3.

**Tabel 3.** Proses training Epoch 40

| Training ke- | Val_accuracy | Val_Loss | Train_accuracy | Train_loss |
|---|---|---|---|---|
| 1 | 97,50% | 0,133 | 99,69% | 0,044 |
| 2 | 96,25% | 0,159 | 99,69% | 0,048 |
| 3 | 98,75% | 0,100 | 99,37% | 0,039 |
| 4 | 97,50% | 0,112 | 98,37% | 0,041 |
| 5 | 98.75% | 0,088 | 99,69% | 0,036 |
| ***Avg*** | **97,50%** | **0,118** | **99,36%** | **0,042** |

Seperti yang ditunjukan pada Tabel 3, proses *training* ke-5 dengan *epoch* 40 menunjukkan nilai yang paling baik di antara hasil *training* lainnya pada *epoch* 40, dengan *train_accuracy* mencapai 99,69% dan *val_accuracy* mencapai 98,75%. Grafik *loss* juga memperlihatkan bahwa *train_loss* menurun drastis mendekati nol, begitu juga dengan *val_loss*. Kemudian untuk rata-rata *val_accuracy* mencapai 97,50% dan rata-rata *train_accuracy* mencapai 99,36%. Perbedaan nilai *loss* yang kecil antara data latih dan validasi menunjukkan bahwa model memiliki stabilitas yang baik selama proses pelatihan.

Tahap terakhir dilakukan dengan menggunakan 50 *epoch* untuk melihat sejauh mana peningkatan jumlah iterasi memengaruhi kemampuan generalisasi model. Peningkatan jumlah *epoch* diharapkan mampu memberikan hasil pelatihan yang lebih stabil dengan tingkat kesalahan yang lebih rendah. Namun, peningkatan ini juga berpotensi menyebabkan model mengalami *overfitting* apabila tidak disertai dengan teknik pengendalian yang tepat. Hasil proses pelatihan pada skenario *epoch* 50 disajikan seperti ditunjukkan pada Tabel 4.

**Tabel 4.** Proses Training Epoch 50

| Training ke- | Val_accuracy | Val_Loss | Train_accuracy | Train_loss |
|---|---|---|---|---|
| 1 | 100,00% | 0,102 | 100,00% | 0,034 |
| 2 | 98,75% | 0,087 | 99,69% | 0,020 |

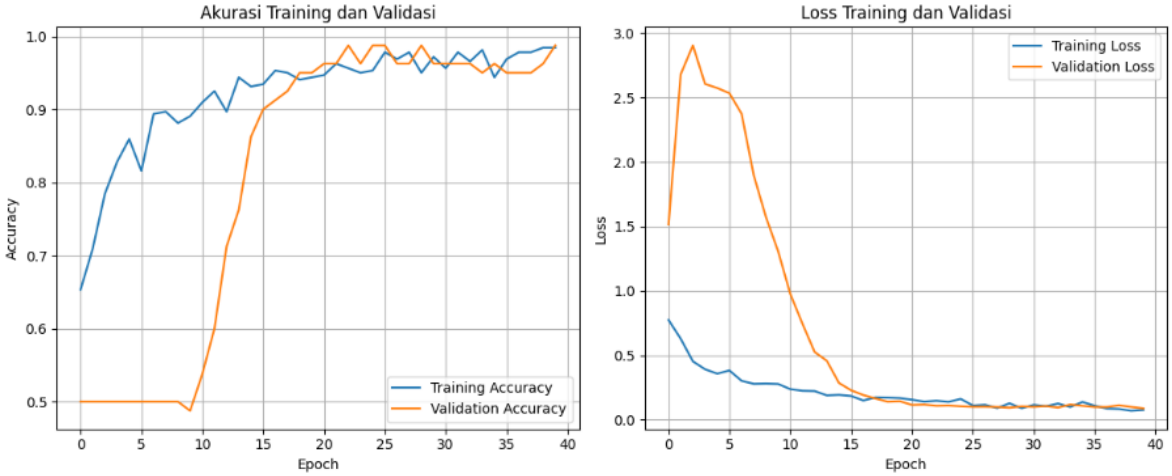
p-ISSN : 2803-1507 e-ISSN : 2803-1531

| | | | | |
|---|---|---|---|---|
| 3 | 96,25% | 0,092 | 100,00% | 0,026 |
| 4 | 97,50% | 0,137 | 100,00% | 0,026 |
| 5 | 97,50% | 0,121 | 99,69% | 0,029 |
| ***Avg*** | **98,00%** | **0,108** | **99,88%** | **0,027** |

Seperti yang ditunjukan pada Tabel 4, proses *training* ke-1 dengan *epoch* 50 menunjukkan nilai yang baik di antara hasil *training* lainnya pada *epoch* 50, dengan *val_accuracy* mencapai 100% dan *train_accuracy* juga mencapai 100%. Model berhasil mengenali pola-pola yang ada dalam *dataset* tanpa melakukan kesalahan prediksi pada data validasi. Namun, meskipun hasil ini terlihat sangat memuaskan, tetap perlu diwaspadai bahwa akurasi yang terlalu sempurna pada data validasi bisa jadi merupakan tanda *overfitting*, di mana model hanya menghafal data yang ada dan gagal melakukan generalisasi terhadap data baru atau data yang belum pernah dilihat sebelumnya. Oleh karena itu, diperlukan evaluasi lebih lanjut menggunakan teknik *regularization* dan *data augmentation* untuk memastikan model tetap memiliki kemampuan generalisasi yang baik.

### 4.3. *Hasil Training Model*

Hasil penelitian ini menunjukkan bahwa penerapan metode *Convolutional Neural Network (CNN)* untuk mendeteksi dan mengklasifikasikan kerusakan jalan berbasis citra digital mampu memberikan performa yang sangat baik. Pada pelatihan dengan *epoch* 50, model mencapai akurasi validasi tertinggi pada *training* ke-1. Hasil pada *epoch* 50 memang menunjukkan akurasi validasi 100% pada sebagian *training*, tetapi jika dilihat dari *val_loss* yang masih relatif tinggi pada sebagian *training*, hal ini mengindikasikan model mulai menghafal data validasi (*overfitting*). Performa pada *epoch* 50 training ke-2 justru lebih stabil karena *val_accuracy* tinggi tanpa perbedaan mencolok dengan *train_accuracy*, dan *val_loss* konsisten rendah. *Epoch* 30 menunjukkan model masih memiliki kelemahan dalam mengenali *non-Cracks*, terlihat dari kesalahan klasifikasi pada sebagian gambar. Hal ini semakin membaik pada *epoch* 40, di mana hampir semua gambar berhasil diklasifikasikan dengan benar. Berdasarkan hasil tersebut, pelatihan terbaik selama uji skenario *epoch* 30, 40, dan 50 diperoleh pada *epoch* 40 *training* ke-5, yang kemudian digunakan sebagai model utama dalam penelitian ini, seperti ditunjukkan pada Gambar 5.



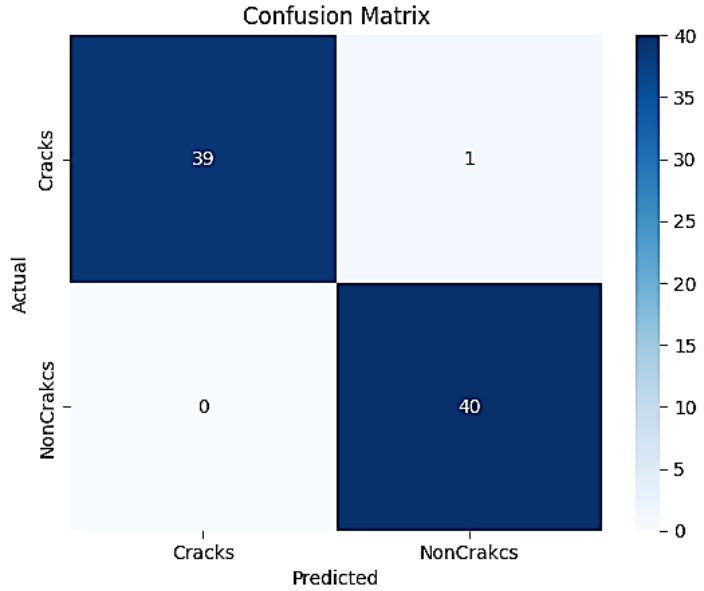

**Gambar 5.** Grafik Hasil Training Ke-5 pada Epoch 40

Pada Gambar 5, ditunjukkan bahwa grafik akurasi memperlihatkan peningkatan signifikan sejak awal pelatihan, di mana akurasi data latih naik dari sekitar 60% hingga lebih dari 90% pada *epoch* ke-20, lalu stabil mendekati 98% hingga akhir pelatihan. Sementara itu, akurasi data validasi yang sempat stagnan di awal pelatihan mulai meningkat tajam sekitar *epoch* ke-10 dan kemudian stabil di atas 95% setelah *epoch* ke-20. Nilai *validation loss* yang relatif rendah menunjukkan bahwa model tidak mengalami penurunan performa yang berarti saat diuji pada data validasi. Hal ini menunjukkan bahwa model mampu belajar secara efektif dari data pelatihan dan mempertahankan performa yang baik tanpa tanda-tanda *overfitting*.

Analisis hasil pelatihan juga diperkuat dengan pemeriksaan *confusion matrix* untuk memahami distribusi prediksi model terhadap kelas *Cracks* dan *non-Cracks*. Informasi ini penting untuk menilai tingkat kesalahan klasifikasi pada setiap kelas dan mengukur kemampuan model dalam mengenali pola yang serupa antar kelas.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Berdasarkan hasil evaluasi, model terbaik diperoleh pada pelatihan ke-5 dengan *epoch* 40, yang menunjukkan keseimbangan performa antara data latih dan validasi, seperti ditunjukkan pada Gambar 6.




**Gambar 6.** Confusion Matrix Training ke-5 Epoch 40

Berdasarkan *confusion matrix* yang ditampilkan pada Gambar 6, model yang dibangun mampu mengklasifikasikan gambar kelas *Cracks* dengan sangat baik, di mana dari 40 gambar, sebanyak 39 gambar berhasil diprediksi dengan benar sebagai *Cracks* (*true positive* = 39), sementara hanya 1 gambar yang salah diklasifikasikan sebagai *non-Cracks* (*false negative* = 1). Sementara itu, untuk kelas *non-Cracks*, model juga menunjukkan performa yang sangat baik dengan 40 gambar berhasil diklasifikasikan dengan benar sebagai *non-Cracks* (*true negative* = 40) dan tidak ada gambar yang salah diklasifikasikan sebagai *Cracks* (*false positive* = 0). Hasil ini mengindikasikan bahwa model telah memiliki kemampuan generalisasi yang tinggi terhadap data uji.

### 4.4. *Perbandingan Dengan Penelitian Sebelumnya*

Jika dibandingkan dengan penelitian oleh [15] yang mana penelitian ini menggunakan *dataset* dari penelitian tersebut, terdapat beberapa perbedaan mendasar. Dalam penelitian ini, peneliti menggunakan *framework* MATLAB dan membangun model CNN dengan tiga lapisan konvolusi, masing-masing memiliki ukuran *filter* 3, 5, dan 7 serta jumlah *filter* sebanyak 8, 16, dan 32. Model mereka menghasilkan akurasi validasi sekitar 94.01%, dengan struktur yang tergolong sederhana dan belum dijelaskan secara rinci adanya penggunaan teknik regularisasi seperti *dropout* atau *batch normalization*. Pendekatan yang digunakan dalam penelitian tersebut masih terbatas pada eksplorasi awal terhadap struktur CNN sederhana. Selain itu, belum terdapat analisis mendalam mengenai pengaruh variasi ukuran *filter* terhadap kemampuan model dalam mengenali pola citra jalan.

**Tabel 5.** Perbandingan Dengan Penelitian Sebelumnya dengan Dataset yang sama

| Aspek | Penelitian Ini | Endri et al. (2020) |
|---|---|---|
| *Framework/Tools* | *TensorFlow & Keras (Python, Google Colab)* | MATLAB |
| Jumlah Data | 400 gambar | 400 gambar |
| Pembagian Data | Manual (*Train* & *Validation*: 80:20) | Tidak dijelaskan |
| Jumlah Lapisan Konvolusi | 4 lapisan konvolusi (*Conv2D*) | 3 lapisan konvolusi |
| Jumlah Filter | 8, 16, 32, 64 | 8, 16, 32 |
| Ukuran Filter | 3x3 pada semua lapisan | 3x3, 5x5, 7x7 |
| Resolusi Gambar | 224x224 piksel | Tidak dijelaskan |
| Regularisasi | *Dropout, BatchNormalization* | Tidak ada |
| Akurasi Validasi Tertinggi | 98.75% (rata-rata sekitar 97.50%) | Sekitar 94.01% |

Sementara itu, pada penelitian ini, model CNN dibangun menggunakan TensorFlow/Keras dalam bahasa pemrograman Python. Model dikembangkan dengan empat lapisan konvolusi yang memiliki *filter* berjumlah 8, 16, 32, dan 64 secara bertahap, memungkinkan jaringan untuk mengenali pola visual yang lebih kompleks

p-ISSN : 2803-1507 e-ISSN : 2803-1531

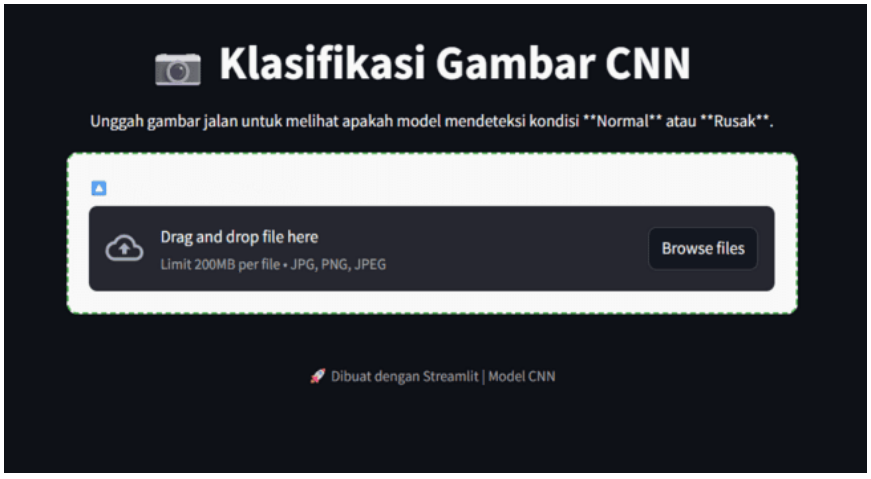
dan berlapis. Selain itu, digunakan fungsi aktivasi ReLU, *batch normalization* untuk setiap lapisan, serta *dropout* sebesar 0.5 pada lapisan *fully connected*. Pendekatan ini memberikan fleksibilitas lebih tinggi dalam proses pelatihan jaringan sehingga model dapat beradaptasi terhadap variasi citra yang lebih luas. Perbandingan karakteristik antara penelitian ini dan penelitian sebelumnya dapat dilihat pada Tabel 5.

Tabel 5, menampilkan perbedaan lebih detail dan jelas. Dalam hal resolusi input, penelitian ini menggunakan ukuran gambar yang lebih kecil, yaitu 224x224 piksel dibandingkan dengan resolusi dalam penelitian sebelumnya. Hasilnya, model dalam penelitian ini mampu mencapai akurasi validasi tertinggi hingga 98.75% dan rata-rata mencapai 97.50%, yang menunjukkan adanya peningkatan performa signifikan dari sisi arsitektur, teknik pelatihan, dan hasil evaluasi. Selain itu, penggunaan ukuran input yang lebih kecil turut mempercepat proses pelatihan tanpa mengurangi akurasi model secara signifikan. Hal ini menunjukkan bahwa optimasi arsitektur jaringan berperan penting dalam efisiensi komputasi serta ketepatan hasil klasifikasi.

Untuk perbandingan lebih lanjut dengan penelitian [9] yang menggunakan algoritma CNN dengan model transfer learning arsitektur VGG16 dan juga ditambahkan penggunaan tahap preprocessing seperti CLACHE dan Gamma Corection, memberikan hasil dengan akurasi sebesar 82% dannilai rata-rata precision sebesar 83% . Selanjutnya juga dalam penelitian dari [11], mengembangkan sistem deteksi secara real-time berbasis algoritma YOLOv11 dengan hasil evaluasi akurasi mencapai 85,34%, nilai rata-rata precision sebesar 75,1%. Kedua penelitian tersebut memperlihatkan variasi pendekatan dalam penerapan CNN untuk deteksi citra jalan. Meskipun demikian, hasil akurasi yang diperoleh masih lebih rendah dibandingkan dengan penelitian ini, yang menunjukkan bahwa desain arsitektur dan teknik *regularization* memberikan pengaruh signifikan terhadap performa model.

### 4.5. *Develop Aplikasi*

Setelah diperoleh model klasifikasi yang paling optimal berdasarkan hasil pelatihan dan evaluasi, langkah penting berikutnya adalah menyimpan model agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang. Model yang telah dilatih disimpan dalam format file (.h5), yang memungkinkan untuk dimuat ulang dan digunakan dalam berbagai aplikasi implementasi, model yang digunakan yakni model pada *training* ke-5 dengan *epoch* 40. Pada penelitian ini, langkah lanjutan yang direncanakan adalah pengembangan aplikasi berbasis web menggunakan *framework* Streamlit. *Framework* ini dipilih karena memungkinkan pembuatan aplikasi web secara cepat dan interaktif dengan memanfaatkan bahasa pemrograman Python, sehingga pengguna cukup mengunggah gambar permukaan jalan ke aplikasi dan mendapatkan hasil klasifikasi secara langsung. Dengan adanya aplikasi ini, model CNN yang telah dibangun dapat diimplementasikan secara praktis dan bermanfaat bagi pengguna umum atau pihak terkait dalam melakukan deteksi kerusakan jalan secara otomatis, cepat, dan efisien. Tampilan antarmuka dari aplikasi yang dikembangkan dapat dilihat pada Gambar 7.



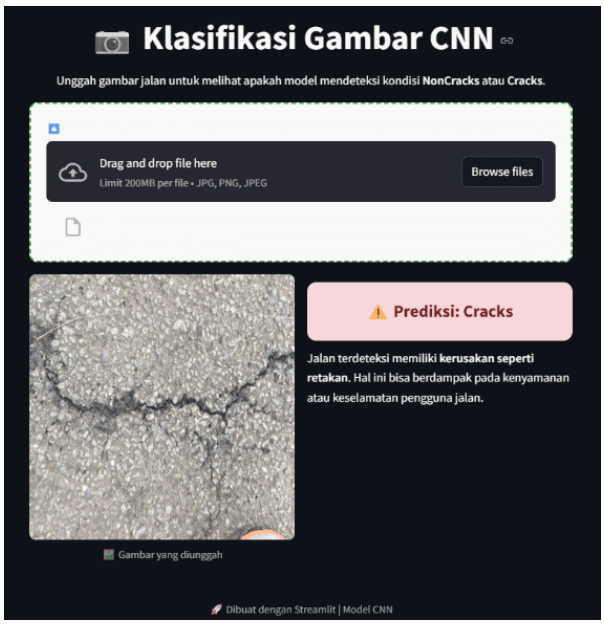

**Gambar 7.** Tampilan Dashboard Website

Pada Gambar 7 ditunjukkan antarmuka aplikasi klasifikasi citra permukaan jalan yang telah dikembangkan menggunakan platform *Streamlit*. Aplikasi ini memungkinkan pengguna untuk mengunggah gambar

p-ISSN : 2803-1507 e-ISSN : 2803-1531

permukaan jalan melalui fitur *upload file* yang tersedia di dalam aplikasi. Dalam contoh pada gambar, pengguna telah berhasil mengunggah citra permukaan jalan yang memperlihatkan kondisi aspal dengan pola retakan yang cukup jelas. Retakan pada permukaan aspal tampak seperti garis-garis pecah yang menyebar di area tertentu pada gambar.

Setelah gambar berhasil diunggah, sistem secara otomatis memproses citra tersebut menggunakan model *Convolutional Neural Network (CNN)* yang telah dilatih sebelumnya. Model CNN kemudian melakukan analisis dan ekstraksi fitur dari citra untuk mengklasifikasikan apakah kondisi jalan tersebut termasuk kategori *Cracks* (jalan retak) atau *non-Cracks* (jalan tidak retak). Pada contoh gambar ini, hasil prediksi yang dihasilkan oleh model menunjukkan label *Cracks*, yang berarti sistem mendeteksi bahwa permukaan jalan pada gambar mengalami kerusakan berupa retakan. Hasil prediksi yang dihasilkan oleh sistem kemudian divisualisasikan dalam bentuk antarmuka hasil seperti ditunjukkan pada Gambar 8.




**Gambar 8.** Tampilan Hasil Prediksi pada Website

Tampilan hasil pada Gambar 8 ini dirancang agar mudah dibaca dan dipahami oleh pengguna. Informasi yang diberikan oleh aplikasi ini diharapkan dapat membantu dalam proses inspeksi kondisi jalan secara otomatis, sehingga mempermudah pengambilan keputusan terkait tindakan perbaikan atau pemeliharaan jalan. Aplikasi ini juga mendemonstrasikan bagaimana teknologi CNN dapat dimanfaatkan dalam sistem pendukung keputusan berbasis citra digital untuk bidang transportasi dan infrastruktur. Desain antarmuka yang sederhana namun informatif menjadikan aplikasi ini mudah digunakan bahkan oleh pengguna yang tidak memiliki latar belakang teknis mendalam.

## 5. DISKUSI

Hasil penelitian menunjukkan bahwa arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan mampu mengenali pola visual pada permukaan jalan dengan akurasi rata-rata 97,50%. Nilai *precision*, *recall*, dan *F1-score* yang tinggi menunjukkan bahwa model memiliki keseimbangan antara kemampuan deteksi dan ketepatan klasifikasi, menandakan generalisasi yang baik terhadap data validasi. Kinerja stabil pada berbagai jumlah *epoch* memperlihatkan bahwa penggunaan *dropout* dan *batch normalization* berhasil mengurangi risiko *overfitting* tanpa menurunkan akurasi. Temuan ini sejalan dengan hasil penelitian [14], [20], [21] yang menekankan efektivitas teknik regularisasi dalam menjaga stabilitas pelatihan pada jaringan konvolusional.

Jika dibandingkan dengan penelitian sebelumnya oleh [15], model CNN yang dikembangkan dalam studi ini menunjukkan peningkatan performa yang signifikan meskipun menggunakan dataset yang sama. Penelitian tersebut menggunakan tiga lapisan konvolusi sederhana dengan akurasi sekitar 94,01%, sedangkan model ini mengadopsi empat lapisan konvolusi dan teknik optimisasi Adam dengan *learning rate* rendah yang menghasilkan akurasi validasi 98,75%. Peningkatan performa ini memperkuat argumen dalam [24], [26] bahwa desain arsitektur dan pemilihan *hyperparameter* berperan besar dalam efisiensi serta akurasi

p-ISSN : 2803-1507 e-ISSN : 2803-1531

klasifikasi citra. Hasil tersebut juga membuktikan bahwa CNN dapat memberikan hasil optimal bahkan tanpa memanfaatkan *transfer learning*, selama struktur jaringan dan parameter pelatihan diatur secara cermat.

Analisis *confusion matrix* menunjukkan bahwa model mampu mengklasifikasikan citra kelas *Cracks* dan *non-Cracks* dengan tingkat kesalahan yang sangat rendah, hanya satu citra yang salah prediksi pada data validasi. Konsistensi hasil ini mengindikasikan bahwa model dapat mengekstraksi fitur tekstur dan pola retakan dengan baik meskipun tanpa tahap *preprocessing* kompleks seperti *CLAHE* atau *Gamma Correction* sebagaimana dilakukan pada penelitian [9]. Meskipun demikian, potensi *overfitting* mulai terlihat pada *epoch* ke-50, ketika nilai *train accuracy* mencapai 100% sementara *validation loss* cenderung meningkat. Fenomena ini menunjukkan perlunya keseimbangan antara jumlah *epoch* dan tingkat regularisasi agar model tidak kehilangan kemampuan generalisasi terhadap data baru.

Implikasi teoretis dari penelitian ini adalah penguatan bukti bahwa CNN efektif dalam mendeteksi kerusakan jalan dengan perbedaan visual yang halus antara retakan dan permukaan aspal utuh. Secara praktis, hasil penelitian mendemonstrasikan potensi penerapan sistem klasifikasi berbasis *deep learning* dalam proses inspeksi infrastruktur secara otomatis untuk menggantikan metode manual yang memakan waktu. Namun, penelitian ini memiliki keterbatasan karena ukuran dataset yang relatif kecil dan belum diterapkan teknik *data augmentation*, sehingga model masih sensitif terhadap variasi pencahayaan dan kondisi permukaan. Selain itu, pengujian model dilakukan hanya pada data statis, sehingga performa sistem dalam kondisi dinamis seperti citra drone atau video lapangan perlu dievaluasi lebih lanjut pada tahap berikutnya.

## 6. KESIMPULAN DAN SARAN

### 6.1. *Kesimpulan*

Berdasarkan hasil penelitian yang telah dilakukan, model *Convolutional Neural Network* (CNN) yang dirancang dan dibangun secara mandiri dari nol, tanpa memanfaatkan arsitektur pra-latih (*pre-trained model*) maupun pendekatan *transfer learning*, mampu mengklasifikasikan kondisi permukaan jalan dengan tingkat akurasi yang sangat baik. Model ini dilatih menggunakan *dataset* yang terbatas yakni berjumlah 400 gambar yang terdiri dari dua kelas seimbang, yaitu 200 gambar kategori *Cracks* (jalan retak) dan 200 gambar kategori non-*Cracks* (jalan tidak retak). Seluruh proses pelatihan dilakukan pada tiga skenario jumlah *epoch*, yaitu 30, 40, dan 50, untuk mengamati konsistensi kinerja model pada berbagai tingkat iterasi pelatihan. Model ini juga diuji secara berulang dengan parameter yang bervariasi untuk memastikan bahwa hasil yang diperoleh tidak hanya bergantung pada konfigurasi pelatihan tertentu.

Dari hasil evaluasi model, diperoleh rata-rata keseluruhan metrik performa, yaitu akurasi sebesar 97,50%, presisi sebesar 96,77%, *recall* sebesar 99,37%, dan *F1-score* sebesar 97,72%. Nilai-nilai ini menunjukkan bahwa model CNN yang dibangun memiliki kemampuan generalisasi yang tinggi dalam mengenali pola visual kerusakan permukaan jalan dari citra digital, baik pada data pelatihan maupun pada data validasi yang belum pernah dilihat sebelumnya. Dengan performa yang konsisten dan tingkat kesalahan yang rendah, sistem ini berpotensi besar untuk diimplementasikan sebagai alat bantu otomatis dalam proses inspeksi jalan, sehingga mampu mempercepat identifikasi kerusakan, mengurangi beban kerja inspeksi manual, dan membantu instansi terkait dalam mengambil keputusan pemeliharaan infrastruktur secara lebih tepat dan efisien. Hasil ini juga menegaskan bahwa arsitektur CNN sederhana pun dapat memberikan performa unggul jika proses pelatihannya dilakukan secara terkontrol dan parameter model diatur dengan tepat.

### 6.2. *Saran*

Beberapa saran dapat diberikan untuk pengembangan penelitian selanjutnya. Pertama, perlu dilakukan pengumpulan *dataset* yang lebih besar dan lebih bervariasi, termasuk menambahkan kategori sub-kerusakan seperti lubang (*potholes*), retakan memanjang, atau *alligator cracks* agar model dapat mengenali jenis kerusakan dengan lebih rinci. Kedua, disarankan untuk meningkatkan teknik *augmentasi data* guna menghasilkan model yang lebih *robust* dan mampu menangani variasi gambar nyata di lapangan, seperti variasi cahaya, sudut pengambilan gambar, dan kondisi cuaca. Ketiga, strategi seperti *early stopping*, regularisasi tambahan, atau pengoptimalan arsitektur model perlu diterapkan untuk mengurangi risiko *overfitting* dan meningkatkan generalisasi model terhadap data baru. Keempat, penelitian selanjutnya sebaiknya mengembangkan aplikasi berbasis web atau *mobile* yang terintegrasi dengan model ini, sehingga dapat digunakan secara langsung oleh pengguna umum atau instansi pemerintah untuk mendeteksi kerusakan jalan secara *real-time*. Terakhir, penting untuk melakukan uji coba lapangan menggunakan gambar yang

p-ISSN : 2803-1507 e-ISSN : 2803-1531

diambil secara langsung dari drone atau kamera ponsel untuk mengukur performa model dalam kondisi nyata dan memastikan kesesuaiannya sebelum diimplementasikan secara luas.

**DAFTAR PUSTAKA**


[1] B. Sasmito, B. H. Setiadji, and R. Isnanto, "Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning di Kota Semarang," *TEKNIK*, vol. 44, no. 1, pp. 7–14, May 2023, doi: 10.14710/teknik.v44i1.51908.

[2] A. W. Mulia, I. Ruslianto, and D. M. Midyanti, "Classification of Road Damage on Highway Images of Pontianak and Surrounding Areas Using Convolutional Neural Network," *Coding: Jurnal Komputer dan Aplikasi*, vol. 11, no. 1, pp. 11–20, 2023. [Online]. Available: https://jurnal.untan.ac.id/index.php/jcskommipa/article/download/57905/75676597138

[3] A. C. Aqsa, H. Mahmudah, and R. W. Sudibyo, "Detection and Classification of Road Damage Using CNN with Hyperparameter Optimization," in *2022 6th International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, Sep. 2022, pp. 101–104. doi: 10.1109/icicos56336.2022.9930607.

[4] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated Road Crack Detection Using Deep Convolutional Neural Networks," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2018, pp. 5212–5215. doi: 10.1109/bigdata.2018.8622327.

[5] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018, doi: 10.1111/mice.12387.

[6] Y. Yulianto and A. Wibowo, "Deteksi Keretakan Jalan Aspal Menggunakan Metode Convolutional Neural Network," *Jurnal Teknik Sipil Cendekia*, vol. 4, no. 2, pp. 581–594, 2023. doi: 10.51988/jtsc.v4i2.132.

[7] E. Setiadi and A. Wibowo, "Klasifikasi dan Deteksi Keretakan Pada Trotoar Menggunakan Metode Convolutional Neural Network," *Jurnal Teknik Sipil Cendekia*, vol. 4, no. 1, pp. 412-427, 2023. doi: 10.51988/jtsc.v4i1.116.

[8] A. Riyandi, T. Widodo, S. Uyun, and U. Islam Negeri Sunan Kalijaga Yogyakarta, "Classification of Damaged Road Images Using the Convolutional Neural Network Method Klasifikasi pada Citra Jalan Rusak Menggunakan Metode Convolutional Neural Network," *Jurnal Informatika dan Teknologi Informasi*, vol. 19, no. 2, pp. 147–158, 2022, doi: 10.31515/telematika.v19i2.6460.

[9] U. Shodiq, *et al*., "Klasifikasi Jalan Rusak Menggunakan Transfer Learning Arsitektur VGG16," *Journal of Information Systems and Informatics Engineering*, vol. 8, no. 1, pp. 75–85, 2024. doi: 10.35145/joisie.v8i1.4243.

[10] T. Takyudin, M. S. Rais, J. A. Putra, and A. Hamsar, "Metode Deteksi dan Estimasi Luas Lubang Jalan Menggunakan Deep Learning Berbasis YOLOv11," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 2, pp. 6786–6790, Jul. 2025, doi: 10.31004/riggs.v4i2.1760.

[11] Y. Triardhana, B. Sasmito, and F. Hadi, "Identifikasi Kerusakan Jalan Menggunakan Metode Deep Learning (DL) Model Convolutional Neural Networks (CNN)," *Jurnal Geodesi Undip*, vol. 10, no. 3, 2021. doi: 10.14710/jgundip.2021.31159.

[12] A. Fajar, D. Iskandar Mulyana, A. Septian Amrullah, K. Irfan Nauval, and T. Informasi Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, "Identifikasi Kerusakan Jalan Dengan Metode Faster R-CNN Studi Kasus Di Jalan Pakansari Bogor Jawa Barat," *Smart Comp : Jurnalnya Orang Pintar Komputer*, vol. 11, no. 2, pp. 247-256, 2022. doi: 10.30591/smartcomp.v11i2.3509.

[13] G. Raudhatul Jannah, Y. Yassir, and H. Hanafi, "Implementasi Deep Learning Dalam Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal TEKTRO*, vol. 8, no. 2, pp. 75-85, 2024. doi: 10.30811/tektro.v8i2.6319.

[14] N. Nafi'iyah, R. Wardhani, and E. Prakasa, "Identification of Banana Ripeness Using Convolutional Neural Network Approaches," in *Proceedings of the 2023 International Conference on Computer,*

p-ISSN : 2803-1507 e-ISSN : 2803-1531

*Control, Informatics and Its Applications (IC3INA)*, Oct. 2023, pp. 330–335. doi: 10.1109/ic3ina60834.2023.10285749

[15] E. Endri, A. Sheta, and H. Turabieh, "Road Damage Detection Utilizing Convolution Neural Network and Principal Component Analysis," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 6, pp. 670–678, 2020. doi: 10.14569/ijacsa.2020.0110682

[16] I. Mulyana and I. Wahyudi, "Deteksi Kerusakan Jalan Berdasarkan Citra Digital Menggunakan Convolutional Neural Network (CNN)," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi (JIMIK)*, vol. 6, no. 1, pp. 294–302, 2025. doi: 10.35870/jimik.v6i1.1192.

[17] A. S. Riyadi, I. P. Wardhani, D. S. Widayati, and K. Kunci, "Klasifikasi Citra Anjing dan Kucing Menggunakan Metode Convolutional Neural Network (CNN)," in *Seminar Nasional Teknologi Informasi dan Komunikasi (SeNTIK)*, vol. 5, no. 1, pp. 307–311, 2021. [Online]. Available: https://ejournal.jak-stik.ac.id/index.php/sentik/article/view/3256

[18] I. Akbar Pradana, A. D. Rahajoe, and A. N. Sihananto, "Pengembangan Aplikasi Pendeteksi Keretakan Jalan Berbasis Android dengan Implementasi Algoritma Hybrid CNN-LSTM," *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 5, no. 2, 2024. doi: 10.33005/jifosi.v5i2.146.

[19] H. Herdianto and D. Nasution, "Klasifikasi Objek Menggunakan Metode Convolutional Neural Network (CNN)," in *Prosiding Seminar Nasional Teknologi Informasi & Komunikasi (SNASTIKOM)*, vol. 1, no. 01, pp. 330–336, 2022. [Online]. Available: https://prosiding.snastikom.com/index.php/snastikom2020/article/view/48

[20] J. Mu, M. Adrezo, and A. N. Haikal, "Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network," *TEKNIKA*, vol. 13, no. 1, pp. 45–50, Jan. 2024, doi: 10.34148/teknika.v13i1.705.

[21] I. G. T. Suryawan and I. P. A. E. D. Udayana, "Optimasi Convolutional Neural Network untuk Deteksi COVID-19 Pada X-Ray Thorax Berbasis Dropout," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 9, p. 551, Jun. 2022, doi: 10.25126/jtiik.202295143.

[22] E. Christianto, A. M. Sambul, and F. D. Kambey, "Implementation of Convolutional Neural Network on Images for Starlings Classification," *Jurnal Teknik Informatika*, 2021. [Online]. Available: https://ejournal.unsrat.ac.id/index.php/informatika

[23] N. Khairunisa, . C., and A. Jamaludin, "Analisis Perbandingan Algoritma CNN Dan YOLO dalam Mengidentifikasi Kerusakan Jalan," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 1756-1768, Aug. 2024, doi: 10.23960/jitet.v12i3.4434.

[24] S. Pariyasto, V. Arfeni Warongan, and Suryani, "Optimalisasi Klasifikasi Citra Medis Menggunakan CNN dan Adam Optimizer dengan Paramater Minimum," *Journal of Science and Social Research*, no. 1, pp. 691–701, 2025, [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR

[25] H. Rohman, N. Nafi'iyah, and A. A. Bettaliyah, "Tomato Ripeness Classification Using Transfer Learning Approach with ResNet50 CNN Model," *Jurnal Ilmiah Sistem Informasi*, vol. 4, no. 3, pp. 646–659, Nov. 2025, doi: 10.51903/zkncwx43.

[26] D. Z. Sholekhah and D. Noviar, "Integrative Deep Learning Architecture for High-Accuracy Medical Image Segmentation: Combining U-Net, ResNet, and Transformers," *Journal of Technology Informatics and Engineering*, vol. 4, no. 1, pp. 115–134, Apr. 2025, doi: 10.51903/jtie.v4i1.288.

[27] S. D. Sriasih, F. A. Razak, and H. al I. Ikhsan, "AI-Driven Sentiment Analysis of Retail Investor Behavior during Market Volatility: A Study of Twitter Data in Southeast Asia," *Journal of Management and Informatics*, vol. 4, no. 1, pp. 741–756, Apr. 2025, doi: 10.51903/jmi.v4i1.179.

[28] M. Fadhilatuzzahro, A. Fazrin, Z. Arishandy Ibtisamah, T. Sukoco Amanda, and H. Maulana, "View of Optimalisasi Model CNN EfficientNet-B0 dengan Fine Tuning untuk Klasifikasi Penyakit Buah Jeruk," *Jurnal Ilmiah Teknologi Informasi dan Robotika*, vol. 7, pp. 23–30, Dec. 2025, doi: https://doi.org/10.33005/jifti.v7i1.162.