

PERBANDINGAN KINERJA K-NEAREST NEIGHBORS DAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI CITRA KONDISI PERMUKAAN JALAN

Fenny Jong¹⁾, Teny Handhayani²⁾

1. Fakultas Teknologi Informasi, Universitas Tarumanagara
email: fenny.535210001@stu.untar.ac.id
2. Fakultas Teknologi Informasi, Universitas Tarumanagara
email: tenyh@fti.untar.ac.id

Abstract

Improving road infrastructure quality is an important aspect of transportation development and road user safety. Automatically assessing road surface conditions can accelerate maintenance and repair efforts. This study compares two classification methods, K-Nearest Neighbors (KNN) and Convolutional Neural Network (CNN), to evaluate road surface conditions based on digital images. Texture features are extracted using the Gray Level Co-occurrence Matrix (GLCM), including Contrast, Homogeneity, Energy, and others, to enhance the classification accuracy in KNN, while feature extraction and classification in CNN are performed automatically. The dataset used in this research consists of 1500 images of road surfaces with three different conditions: smooth, cracked, and potholes. Each condition contains 500 images with a resolution of 300x300 pixels. The results show that the KNN algorithm achieves an accuracy of 57.2%, while CNN demonstrates the best performance with an accuracy of 93.8% for 80% training data and 20% testing data

Kata Kunci : Convolutional Neural Network, Road Infrastructure, Gray Level Co-occurrence Matrix, K-Nearest Neighbors, Classification.

A. PENDAHULUAN

Peningkatan kualitas infrastruktur jalan merupakan aspek penting dalam pembangunan transportasi dan keselamatan pengguna jalan. Permukaan jalan memiliki peran krusial dalam mendukung kelancaran mobilitas serta aktivitas masyarakat sehari-hari [1]. Jalan yang baik tidak hanya meningkatkan efisiensi perjalanan, tetapi juga berpengaruh pada keselamatan dan kenyamanan pengguna. Kondisi permukaan jalan yang baik dapat memperpanjang umur pakai jalan tersebut dan mengurangi potensi risiko

kecelakaan. Oleh karena itu, pemantauan dan klasifikasi kondisi permukaan jalan secara berkala sangat penting untuk memastikan kualitas jalan tetap terjaga. Oleh karena itu, pemantauan dan klasifikasi kondisi permukaan jalan menjadi hal yang penting untuk dilakukan secara berkala [2]. Pengecekan kualitas jalan raya biasanya dilakukan secara manual yang memerlukan banyak waktu dan tenaga. Jumlah jalan yang banyak dan proses pengecekan manual merupakan masalah utama dalam perawatan jalan raya [3]. Salah satu cara untuk mendeteksi dan menganalisis kondisi jalan adalah

dengan menggunakan teknik pembelajaran mesin untuk mengklasifikasikan gambar atau citra permukaan jalan.

Hubungan spasial antar piksel diamati oleh ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) yang kemudian diklasifikasikan dengan *K-Nearest Neighbors* (K-NN) dan *Convolutional Neural Network* (CNN) merupakan dua algoritma pembelajaran mesin yang populer dan sering digunakan dalam tugas klasifikasi, termasuk klasifikasi citra. K-NN merupakan metode pembelajaran berbasis *instance* yang sederhana namun efektif dalam berbagai masalah klasifikasi [4][5], sementara CNN sebagai salah satu bentuk arsitektur jaringan saraf dalam, terbukti sangat unggul dalam pemrosesan dan pengenalan citra berkat kemampuannya dalam mengekstraksi fitur secara otomatis [6].

Penelitian sebelumnya telah banyak membahas penggunaan algoritma K-NN dan CNN untuk berbagai aplikasi klasifikasi citra, seperti pengenalan objek, pengklasifikasian citra medis, hingga analisis citra jalan. Namun, perbandingan kinerja kedua algoritma ini dalam klasifikasi kondisi permukaan jalan masih terbatas. Beberapa penelitian menunjukkan bahwa CNN lebih unggul dalam hal akurasi dan kemampuan generalisasi dibandingkan metode lain, namun metode yang lebih sederhana seperti K-NN tetap relevan dalam situasi tertentu dengan dataset yang terbatas atau ketika interpretabilitas model sangat penting. Misalnya, Penelitian Ridol Liusman dkk yang mengembangkan model CNN berbasis arsitektur *Xception* untuk klasifikasi kualitas permukaan jalan, dengan hasil akurasi mencapai 90,11% [7]. Penelitian oleh Budiarto dkk. mengenai sistem klasifikasi permukaan jalan dan penghindaran jalan berlubang

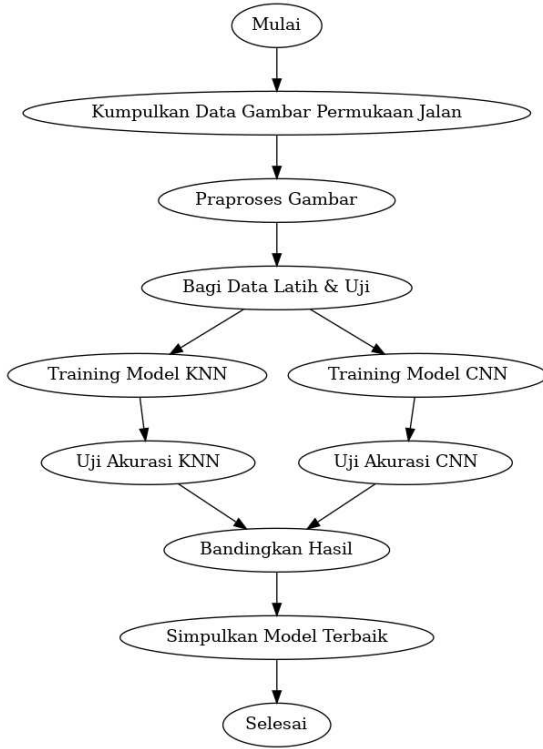
pada kursi roda pintar menggunakan metode *MobileNetV3-Small* mencapai akurasi 88,076% dengan waktu komputasi rata-rata 0,96048 detik [7]. Selain itu, Penelitian oleh Muchtar dkk. berjudul penerapan analisis berbasis fraktal dalam klasifikasi citra retakan pada permukaan jembatan beton menggunakan *K-Nearest Neighbor* (K-NN) berhasil mencapai akurasi 97,6%, dengan presisi 97,659% dan recall 97,6% [8]. Leboh versi 2 merupakan aplikasi deteksi sampah yang menerapkan metode *Efficient-Det* memperoleh akurasi 82% [9]. Aplikasi untuk klasifikasi rumah adat Indonesia menerapkan metode *Efficient-Net* memperoleh akurasi 91% [10]. *InceptionV3* dan *Xception* berhasil digunakan untuk klasifikasi bangunan Pura di Bali [11]. *MobileNetv2* dan *InceptionV3* mendapatkan akurasi lebih dari 90% untuk mengklasifikasikan jenis buah [12].

Penelitian ini bertujuan untuk membandingkan algoritma KNN dan CNN dalam klasifikasi kondisi permukaan jalan. Diharapkan metode ini akan berfungsi sebagai dasar untuk pengembangan sistem pemantauan kondisi jalan yang menggunakan pembelajaran mesin yang lebih efisien.

B. METODE PENELITIAN

Tahapan proses pengembangan model untuk pengenalan gambar permukaan jalan digambarkan dalam *flowchart* yang ditunjukkan pada Gambar 1. Dimulai dengan data gambar permukaan jalan dikumpulkan, yang kemudian diproses melalui tahap pra-proses gambar. Selanjutnya, dikelompokkan data latihan dan data pengujian. Model *K-Nearest Neighbors* (KNN) dan *Convolutional Neural Network* (CNN) kemudian dilatih dengan data latih dan kemudian diuji akurasi

dengan data pengujian. Hasil dari kedua model dievaluasi sebelum memilih model yang lebih baik dan pada akhirnya, model yang paling akurat diputuskan. Setelah itu, proses diakhiri.



Gambar 1. Flowchart Penelitian

Gray-Level Co-occurrence Matrix (GLCM) adalah metode ekstraksi fitur untuk analisis tekstur yang menghitung probabilitas hubungan antara dua piksel berdasarkan intensitas, jarak, dan orientasi sudut (0° , 45° , 90° , 135°) dengan jarak antar piksel 1. Tahapannya meliputi pembentukan matriks awal, penggabungan matriks, perhitungan probabilitas elemen, dan ekstraksi fitur dari semua arah sudut. Persamaan rumus GLCM sebagai berikut [13]:

$$Contrast = \sum_{i,j=0}^{N-1} P_{(ij)} (i - j)^2 \quad (1)$$

$$Energy = \sum_{i,j=0}^{N-1} (P_{(ij)})^2 \quad (2)$$

$$Entropy = \sum_{i,j=0}^{N-1} -\ln(P_{(ij)}) P_{(ij)} \quad (3)$$

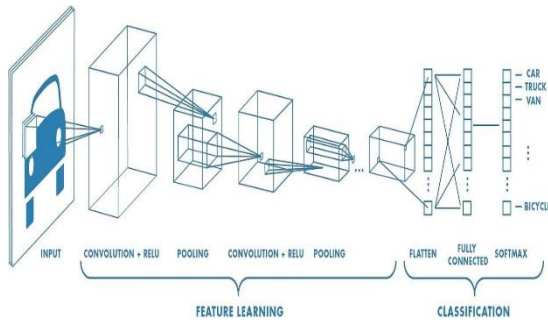
$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{(ij)}}{(1+(j-i)^2)} \quad (4)$$

$$correlation = \frac{\sum_i \sum_j (ij) P(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (5)$$

K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin untuk regresi dan klasifikasi. KNN mengklasifikasikan titik data menggunakan K tetangga terdekat dan mayoritas kelas tetangga. Algoritma ini menentukan kedekatan titik data dengan menggunakan jarak antara mereka. Ada persamaan berikut untuk rumus KNN [13] [14], di mana d adalah jarak data, i adalah variabel data, n adalah dimensi data A_i adalah data latih B_i adalah data pengujian.

$$d(A, B) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2} \quad (6)$$

Convolutional Neural Network (CNN) adalah metode populer untuk pengolahan citra karena kemampuannya mengolah data gambar besar dengan akurasi tinggi dan efisiensi proses. CNN unggul dalam menyimpan informasi spasial, dengan struktur yang mencakup input, hidden layer, klasifikasi, dan output, serta menggunakan fungsi aktivasi seperti Softmax untuk menghasilkan klasifikasi [15][16]. Gambar 2 adalah visualisasi CNN.



Gambar 2. Convolutional Neural Network (CNN)

Berikut ini penjelasan tiap struktur CNN:

- a. *Input Layer*
Lapisan input adalah lapisan pertama yang menyimpan nilai piksel gambar yang dimasukkan [17].
- b. *Convolution Layer*
Lapisan ini mengekstrak fitur penting dari citra dengan operasi konvolusi menggunakan filter. Setiap convolutional layer mengidentifikasi fitur sederhana pada lapisan awal dan fitur lebih kompleks pada lapisan dalam. Rumus *Convolution Layer* sebagai berikut [17]:

$$(of)_{ij} = (w_f * R)_{ij} + b_{ij} \quad (7)$$

Pada persamaan ini, o_{ij} adalah nilai keluaran di posisi i, j yang diperoleh dari operasi konvolusi antara filter w_j dan input R , menghasilkan $(w_f * R)_{ij}$. Nilai b_{ij} kemudian ditambahkan untuk meningkatkan kemampuan model menangkap informasi kompleks.

- c. *Activation Layer*
Fungsi aktivasi menentukan apakah sinyal diteruskan ke lapisan berikutnya. Fungsi umum yang digunakan ReLU yaitu mengubah nilai input negatif menjadi nol, mempercepat komputasi, dan mengurangi risiko *overfitting* [18]. Sigmoid yaitu fungsi non-linear yang menghasilkan nilai antara 0 dan 1, sering digunakan untuk klasifikasi

biner [19]. *Softmax* yaitu menghitung probabilitas dan menentukan kelas berdasarkan nilai yang dinormalisasi.

- d. *Pooling Layer*
Pooling layer digunakan untuk mengurangi *feature map* konvolusi, mempercepat komputasi, dan mengekstraksi fitur dominan [20].
- e. *Flatten*
Proses ini mengubah *feature map* menjadi vektor yang berdimensi satu untuk memudahkan klasifikasi [21].
- f. *Fully Connected Layer*
Setelah proses convolution dan pooling hasilnya kemudian diubah menjadi vektor input untuk *fully connected layer* yang setiap nilai dalam vektor ini mewakili probabilitas untuk kelas tertentu dari gambar yang diklasifikasikan [22].

Metode evaluasi digunakan untuk menilai kinerja model dalam tugas pembelajaran mesin. Beberapa metrik evaluasi klasifikasi yang digunakan antara lain akurasi, precision, recall, dan F1-score. Metode ini membantu dalam memahami seberapa baik model dalam memprediksi hasil yang benar dan menghindari kesalahan. *Confusion Matrix* menggambarkan jumlah prediksi untuk setiap kombinasi kelas aktual dan kelas prediksi (*True Positive*, *True Negative*, *False Positive*, *False Negative*) [23] [24] [24]. Gambar 3 adalah visualisasi confusion matrix.

	Prediction outcome		
	positive	negative	
positive	<i>TP</i>	<i>FN</i>	$TP + FN$
negative	<i>FP</i>	<i>TN</i>	$FP + TN$
	$TP + FP$	$FN + TN$	

Gambar 3. Confusion Matrix

Precision adalah ukuran jumlah prediksi positif yang benar-benar relevan. Kemampuan model untuk menemukan semua kasus positif diukur dengan *recall*. Rumus *recall* sebagai berikut. Akurasi mengukur persentase prediksi yang benar dari total data. F1-Score adalah skor rata-rata untuk presisi dan *recall* yang mengimbangi keduanya. Rumus *precision*, *recall*, *f1-score*, dan akurasi dihitung dengan rumus berikut [25]:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$precision = \frac{TP}{TP + FP} \quad (9)$$

$$recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 - Score = \frac{2 * Precision * recall}{Precision + recall} \quad (11)$$

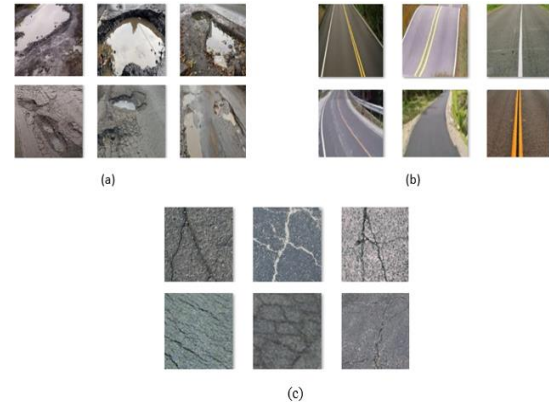
C. HASIL DAN PEMBAHASAN

Dataset dalam penelitian ini diperoleh melalui *web scraping* dari berbagai sumber *online* yang menyediakan gambar kondisi permukaan jalan. *Dataset* terdiri dari 500 gambar per kelas, meliputi kondisi jalan mulus, retak, dan berlubang, yang digunakan untuk klasifikasi. Penjelasan lebih lanjut tentang kelas-kelas ini terdapat pada Tabel 1, sementara contoh gambar dataset disajikan pada Gambar 4.

Tabel 1. Kelas Kondisi Permukaan Jalan

Kelas	Deksripsi
Jalan Mulus	Gambar yang menunjukkan permukaan jalan yang rata dan tidak terdapat kerusakan atau lubang.
Jalan Retak	Gambar yang menggambarkan jalan dengan kerusakan

	permukaan seperti retakan atau degradasi ringan.
Jalan Berlubang	Gambar yang memperlihatkan jalan dengan lubang besar atau kerusakan yang cukup parah.



Gambar 4. Contoh Dataset (a) Jalan Berlubang, (b) Jalan Mulus, dan (c) Jalan Retak.

Tahapan *preprocessing* dilakukan untuk memastikan data sesuai kebutuhan model. Semua gambar diubah ukurannya menjadi 300x300 piksel untuk menyamakan dimensi dan mengurangi kebutuhan memori tanpa kehilangan detail penting. *Dataset* dibagi dalam dua skenario pelatihan dan pengujian: 80:20 dan 70:30. Pada skenario 80:20, 80% dialokasikan untuk dilatih dan 20% untuk pengujian. Sedangkan pada skenario 70:30, 70% data dialokasikan untuk dilatih dan 30% untuk pengujian.

Pengujian *K-Nearest Neighbors* (KNN) dilakukan dengan $k = 3$ dan berbagai nilai random state (1, 10, 100, 1000, dan 10.000). Selain itu, data pelatihan dan pengujian dibagi menjadi *dataset* 80:20 dan 70:30 untuk mendapatkan informasi tekstual dari gambar, *Gray-Level Co-occurrence Matrix* (GLCM) digunakan untuk

mengeksktraksi fitur. Konfigurasi KNN dengan pembagian dataset 70% data Latihan dan 30% data pengujian ditunjukkan dalam Tabel 3 dan Tabel 2 menunjukkan 80% data latihan dan 20% data pengujian.

Tabel 2a. Hasil Pengujian KNN 80% Data Latihan dan Data pengujian 20%

Random State	Precision	Recall
1	0.57	0.56
10	0.58	0.58
100	0.58	0.58
1000	0.57	0.58
10000	0.57	0.56

Tabel 2b. Hasil Pengujian KNN 80% Data Latihan dan Data pengujian 20%

F1-score	Akurasi	Waktu Pelatihan (detik)
0.55	0.56	0.0039
0.58	0.59	0.0000
0.58	0.59	0.0105
0.57	0.56	0.0028
0.56	0.56	0.0021

Pembagian dataset 80:20 pada Tabel 2 menunjukkan kinerja yang cukup stabil dengan nilai *precision* dan *recall* berkisar antara 0.56 hingga 0.58, dan nilai tertinggi tercatat pada *random state* 10 dan 100, masing-masing mencapai 0.58. *F1-score* juga bervariasi antara 0.55 hingga 0.58, dengan performa terbaik pada *random state* 10 dan 100. Sementara itu, akurasi pada pengujian ini berkisar antara 0.56 hingga 0.59, dengan random state 10 dan 100 mencatatkan nilai akurasi tertinggi sebesar 0.59.

Tabel 3a. Hasil Pengujian KNN 70% Data Latihan dan Data pengujian 30%

Random State	Precision	Recall
1	0.58	0.56
10	0.58	0.58
100	0.55	0.55
1000	0.57	0.57
10000	0.58	0.58

Tabel 3b. Hasil Pengujian KNN 70% Data Latihan dan Data pengujian 30%

F1-score	Akurasi	Waktu Pelatihan (detik)
0.56	0.56	0.0012
0.57	0.58	0.0000
0.55	0.55	0.0000
0.57	0.56	0.0000
0.58	0.58	0.0041

Pembagian *dataset* 70:30 pada Tabel 3 menunjukkan hasilnya sedikit lebih rendah dibandingkan dengan pembagian 80:20, namun masih menunjukkan kinerja yang relatif stabil. *Precision* dan *recall* berada dalam kisaran 0.55 hingga 0.58, dengan nilai terbaik tercatat pada *random state* 10 dan 10000, yang mencapai 0.58. Begitu juga dengan *F1-score*, yang berkisar antara 0.55 hingga 0.58, dengan performa tertinggi pada *random state* 10 dan 10000. Akurasi pada pengujian ini berada di kisaran 0.55 hingga 0.58, dengan performa terbaik mencapai 0.58 pada *random state* 10 dan 10000.

Pengujian *Convolutional Neural Network* (CNN) dengan ResNet152V2 dilakukan dengan berbagai jumlah *epoch* (10, 20, 30, 40, dan 50). Tujuan dari eksperimen adalah untuk mengevaluasi pengaruh jumlah *epoch* terhadap kinerja model dalam mengklasifikasikan kondisi

permukaan jalan. Diharapkan kemampuan model untuk mendeteksi fitur gambar dengan lebih baik dapat ditingkatkan dengan meningkatkan jumlah *epoch*.

Konfigurasi CNN dengan pembagian data 80% data latihan dan 20% data pengujian ditunjukkan dalam Tabel 4 dan konfigurasi CNN dengan pembagian data 70% data latihan dan 30% data pengujian ditunjukkan di Tabel 5. Pembagian 80:20 menghasilkan jumlah 38 untuk setiap langkah *epoch*, sementara pembagian 70:30 menghasilkan jumlah 33 untuk setiap langkah *epoch*, memberikan gambaran tentang perbedaan dalam distribusi data antara kedua pengaturan tersebut.

Tabel 4a. Hasil Pengujian CNN 80% Data Latihan dan Data pengujian 20%

Epoch	Precision	Recall
10	0.94	0.93
20	0.94	0.94
30	0.94	0.93
40	0.96	0.96
50	0.94	0.93

Tabel 4b. Hasil Pengujian CNN 80% Data Latihan dan Data pengujian 20%

F1-score	Akurasi	Waktu pelatihan (detik)
0.93	0.93	962
0.94	0.94	1547
0.93	0.93	1897
0.96	0.96	2705
0.93	0.93	3054

Tabel 4 menunjukkan hasil pengujian model CNN dengan pembagian *dataset* 80% untuk latihan dan 20% untuk pengujian. Hasil pengujian menunjukkan seiring bertambahnya *epoch*, nilai *Precision*, *Recall*, *F1-score*, dan akurasi cenderung stabil pada angka yang tinggi,

dengan nilai tertinggi dicapai pada *epoch* ke-40 yakni 0.96 untuk *Precision*, *Recall*, *F1-score*, dan akurasi, meskipun waktu pelatihan juga semakin lama.

Tabel 5a. Hasil Pengujian CNN 70% Data Latihan dan Data pengujian 30%

Epoch	Precision	Recall
10	0.91	0.91
20	0.94	0.94
30	0.93	0.93
40	0.94	0.94
50	0.93	0.93

Tabel 5b. Hasil Pengujian CNN 70% Data Latihan dan Data pengujian 30%

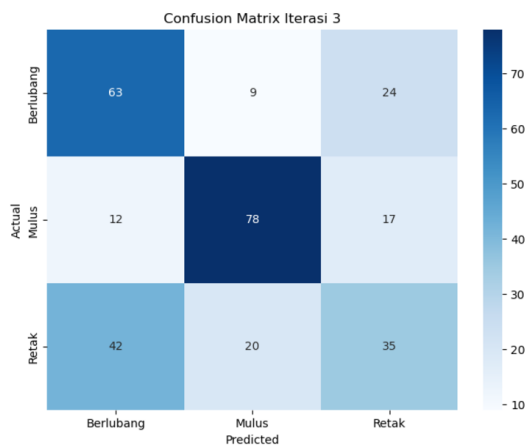
F1-score	Akurasi	Waktu pelatihan (detik)
0.91	0.91	633
0.94	0.94	1183
0.93	0.93	1870
0.94	0.94	2733
0.93	0.93	3276

Tabel 5 menunjukkan hasil pengujian dengan 70% untuk latihan dan 30% untuk pengujian. Meskipun hasil *Precision*, *Recall*, *F1-score*, dan akurasi sedikit lebih rendah dibandingkan dengan pembagian 80%:20%, model ini tetap menunjukkan performa yang cukup baik dengan nilai tertinggi pada *epoch* ke-20 yakni 0.94 untuk *Precision*, *Recall*, *F1-score*, dan akurasi. Waktu pelatihan juga lebih cepat dibandingkan dengan pembagian 80%:20%, terutama pada *epoch* pertama yang memerlukan waktu hanya 633 detik.

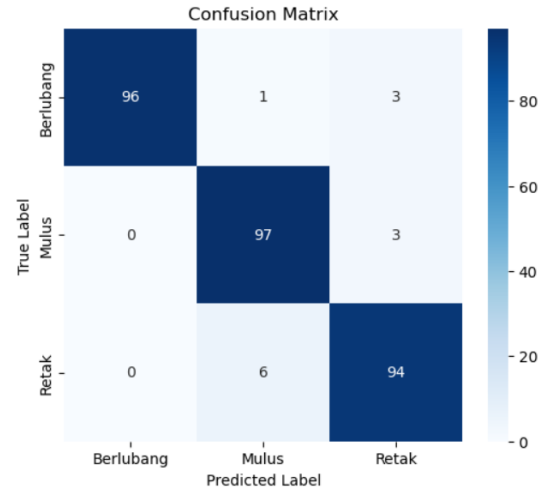
Eksperimen ini mengevaluasi perbandingan dua metode klasifikasi yang digunakan untuk mengklasifikasikan kondisi permukaan jalan, yaitu *K-Nearest Neighbors* (KNN) dan *Convolutional Neural Network* (CNN). *Confusion matrix*

digunakan untuk menampilkan performa klasifikasi dari kedua model, yang memvisualisasikan prediksi benar dan salah untuk masing-masing kelas yaitu berlubang, mulus, dan retak.

Gambar 5 menunjukkan *confusion matrix* dari model *K-Nearest Neighbors* (KNN) dengan pembagian data latih 80% dan data pengujian 20%. Model ini memiliki akurasi rendah, terutama pada data dengan label retak, yang sering salah diprediksi sebagai berlubang menunjukkan keterbatasan KNN dalam menangani klasifikasi berbasis citra kompleks. Sebaliknya Gambar 6 menampilkan *confusion matrix* dari model *Convolutional Neural Network* (CNN) yang memberikan performa terbaik dengan pembagian data yang sama. CNN mampu mengklasifikasikan data dengan tingkat akurasi tinggi, dengan prediksi benar dominan pada kategori berlubang (96), mulus (97), dan retak (94). Hal ini menunjukkan keunggulan CNN dalam menangkap pola visual kompleks, menjadikannya metode yang lebih andal dibandingkan KNN.



Gambar 5. Confusion Matrix KNN Pembagian 80% Data Latihan dan 20% Data pengujian



Gambar 6. Confusion Matrix CNN Pembagian 80% Data Latihan dan 20% Data pengujian

Tabel 6a. Rata-rata Hasil Pengujian

Algoritma	Data Latih	Data pengujian	Rata-rata Precisi on
KNN (k=3)	80	20	0.574
KNN (k=3)	70	30	0.572
CNN (ResNet152 V2)	80	20	0.944
CNN (ResNet152 V2)	70	30	0.93

Tabel 6b. Rata-rata Hasil Pengujian

Rata-rata Recall	Rata-rata F1-score	Rata-rata Akurasi	Rata-rata Waktu pelatihan (detik)
0.572	0.568	0.572	0.00386
0.568	0.566	0.566	0.00106
0.938	0.938	0.938	2033
0.93	0.93	0.93	1939

dibandingkan dengan KNN. CNN menghasilkan rata-rata *Precision*, *Recall*, *F1-score*, dan akurasi masing masing

sebesar 0.944, 0.938, 0.938, dan 0.938 dengan waktu pelatihan rata-rata 2033 untuk 80% latihan dan 20% uji. Sebaliknya, KNN dengan $k = 3$ hanya untuk pembagian dataset 80% latihan dan 20% uji hanya memperoleh nilai rata-rata *Precision*, *Recall*, *F1-score*, dan akurasi masing masing sebesar 0.574, 0.572, 0.568, dan 0.572, dengan waktu pelatihan yang sangat cepat yaitu hanya 0.00386 detik. Demikian pula, pada pembagian dataset 70% latih dan 30% uji, CNN tetap menunjukkan performa yang lebih unggul dengan rata-rata *Precision*, *Recall*, *F1-score*, dan akurasi 0.93, dibandingkan KNN yang hanya mencapai 0.572 untuk *Precision* dan 0.566 untuk *F1-score*. Meskipun KNN memerlukan waktu pelatihan yang lebih singkat, CNN menunjukkan keunggulan yang signifikan dalam akurasi dan kinerja klasifikasinya pada pembagian dataset 80% latihan dan 20% pengujian. Pada pembagian data ini, menghasilkan nilai rata-rata *Precision* sebesar 0.944, *Recall* sebesar 0.938, *F1-score* sebesar 0.938, dan akurasi sebesar 0.938.

D. KESIMPULAN DAN SARAN

Eksperimen ini menunjukkan bahwa Convolutional Neural Network (CNN) memiliki akurasi sebesar 93,8%, sedangkan K-Nearest Neighbors (KNN) memiliki akurasi 57,8%. CNN menunjukkan prediksi yang lebih akurat pada semua kelas terutama pada kondisi retak dengan lebih sedikit kesalahan klasifikasi dibandingkan KNN. Prediksi CNN konsisten mengenali pola visual kompleks, sementara KNN cenderung menghasilkan kesalahan lebih tinggi terutama pada kelas retak dan berlubang. Hasil ini menegaskan bahwa CNN lebih unggul untuk klasifikasi kondisi

permukaan jalan karena kemampuannya dalam ekstraksi fitur otomatis, meskipun membutuhkan waktu pelatihan yang lebih lama. Sebaliknya, KNN lebih sederhana namun kurang efektif dalam menangani pola yang kompleks.

Pengembangan selanjutnya dapat dilakukan dengan menggunakan arsitektur CNN yang lebih canggih untuk meningkatkan akurasi dan efisiensi model. Selain itu, augmentasi data dapat digunakan untuk meningkatkan variasi data latih agar model lebih andal. Pendekatan hybrid, seperti mengombinasikan CNN untuk ekstraksi fitur dan model KNN atau SVM sebagai pengklasifikasi juga dapat menjadi solusi untuk mengoptimalkan kelebihan kedua metode. Evaluasi pada data jalan yang lebih beragam juga diperlukan untuk memastikan generalisasi model di kondisi nyata. Dengan pengembangan ini, sistem klasifikasi kondisi permukaan jalan dapat menjadi lebih akurat dan efektif untuk diterapkan dalam sistem pemantauan jalan otomatis.

E. REFERENSI

- [1] Y. Sari, A. R. Baskara, P. B. Prakoso and M. A. Rahman, "Penerapan Active Contour Model pada Pengolahan Citra untuk Deteksi Kerusakan Jalan," *Jurnal Jalan dan Jembatan*, vol. 38, no. 2, pp. 138 - 147, 2021.
- [2] R. L. Gaho, I. T. Ali and E. Prakasa, "Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur Xception," *Jurnal Inovtek Polbeng Seri Informatika*, vol. 9, no. 1, pp. 354-365, 2024.

- [3] A. Wibowo and E. Setiyadi, "Klasifikasi dan Deteksi Keretakan Pada Trotoar Menggunakan Metode Convolutional Neural Network," *Jurnal Teknik Sipil Cendekia*, vol. 4, no. 1, pp. 411-425, 2023.
- [4] I. Maylani, V. W. Ambarwati, B. Wasykuru, A. Alqaroni and F. T. B. K. Wati, "Grey Level Co-Occurrence Matrix (GLCM) & Hybrid Klasifikasi untuk Mendeteksi Kerusakan Jalan Aspal," in *Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika*, Surabaya, 2023.
- [5] S. Ulya, M. A. Soeleman and F. Budiman, "Optimasi Parameter K Pada Algoritma K-NN Untuk Klasifikasi Prioritas Bantuan Pembangunan Desa," *Techno.COM*, vol. 20, no. 1, pp. 83-96, 2021.
- [6] M. H. Fadlun, M. Martanto and U. Hayati, "Klasifikasi Tumor Otak menggunakan Convolutional Neural Network dan Transfer Learning," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 6, no. 1, p. 289-295, 2024.
- [7] M. R. W. Budiarto and F. Utaminigrum, "Sistem Klasifikasi Permukaan Jalan dan Penghindaran Jalan Berlubang pada Kursi Roda Pintar dengan Metode MobileNetV3-Small," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 5, p. 2249-2254, 2023.
- [8] M. Muchtar, M. N. Sutoyo, A. Paliling, S. Sunyanti and J. N. Iin, "Penerapan Analisis Berbasis Fraktal dalam Klasifikasi Citra Retakan pada Permukaan Jembatan Beton," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 9, no. 1, p. 21-29, 2024.
- [9] T. Handhayani, A. H. Pawening and J. Hendryli, "Leboh 2: An Android Application for Solid Waste," *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1-9, 2023.
- [10] T. Handhayani, A. H. Pawening and J. Hendryli, "An Automatic Recognition System for Digital Collections of Indonesian Traditional Houses Using Convolutional Neural Networks for Cultural Heritage Preservation," *International Journal of Computational Intelligence and Applications*, vol. 22, no. 2, pp. 1-13, 2023.
- [11] E. Jochsen and T. Handhayani, "Pengenal Bangunan Bersejarah Pura Dengan Arsitektur InceptionV3 dan Xception," *Jurnal Eksplora Informatika*, vol. 14, no. 1, pp. 1-11, 2024.
- [12] B. Karnadi and T. Handhayani, "Klasifikasi Jenis Buah dengan Menggunakan Metode MobileNetv2 dan Inceptionv3," *Jurnal Eksplora Informatika*, vol. 14, no. 2, pp. 35-42, 2024.
- [13] R. A. Saputra, S. Suharyanto, S. Wasiyanti, D. F. Saefudin, A. Supriyatna and A. Wibowo, "Rice Leaf Disease Image Classifications Using KNN Based On GLCM Feature Extraction," in *International Conference on Advanced Information Scientific Development (ICAISD)*, 2020.
- [14] S. R. Cholil, T. Handayani, R. Prathivi and T. Ardianita, "Implementasi Algoritma

- Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa,” *IJCIT (Indonesian Journal on Computer and Information Technolog*, vol. 6, no. 2, pp. 118-127, 2021.
- [15] Y. LeCun, Y. Bengio and G. Hinton, “Deep Learning,” *Nature*, vol. 521, p. 436–444, 2015.
- [16] T. Handhayani and J. Hendryli, “Leboh: An Android Mobile Application for Waste Classification Using TensorFlow Lite,” in *Intelligent Systems Conference (IntelliSys)*, Amsterdam, 2022.
- [17] K. Azmi, S. Defit and S. Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” *Jurnal Unitek*, vol. 16, no. 1, pp. 28-40, 2023.
- [18] S. Sikandar, R. Mahum and A. Alsalman, “A Novel Hybrid Approach for a Content-Based Image Retrieval Using Feature Fusion,” *Applied Sciences*, vol. 13, no. 7, pp. 1-17, 2023.
- [19] T. L. Malau, S. Sawaluddin, S. Sutarman and T. Joseph, “Analisis Metode Logistik Regresi Ensemble untuk Klasifikasi dengan Pra-Pemrosesan Menggunakan Principal Component Analysis,” *IJM: Indonesian Journal of Multidisciplinary*, vol. 1, no. 2, pp. 707-722, 2023.
- [20] M. R. Alwanda, R. P. K. Ramadhan and D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle,” *Jurnal Algoritme*, vol. 1, no. 1, p. 45–56, 2020.
- [21] D. R. R. Putra and R. A. Saputra, “Implementasi Convolutional Neural Network (CNN) untuk Mendeteksi Penggunaan Masker pada Gambar,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3, 2023.
- [22] K. R. Wardani and L. Leonardi, “Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network,” *Jurnal Tekno Insentif*, vol. 17, no. 2, pp. 112-126, 2023.
- [23] B. P. Pratiwi, A. S. Handayani and S. Sarjana, “Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi WSN Menggunakan Confusion Matrix,” *urnal Informatika Upgris*, vol. 6, no. 2, 2020.
- [24] R. R. Adhitya, W. Witanti and R. Yuniarti, “Perbandingan Metode CART dan Naïve Bayes untuk Klasifikasi Customer Churn,” *INFOTECH journal*, vol. 9, no. 2, p. 307–318, 2023.
- [25] B. V. Jayadi, M. D. Lauro, Z. Rusdi and T. Handhayani, “Klasifikasi Indeks Standar Pencemaran Udara untuk Data Tidak Seimbang menggunakan Pendekatan Pembelajaran Mesin,” *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no. 3, pp. 951-958, 2024.