

Sistem Deteksi Pothole Menggunakan Convolutional Neural Network Dengan Squeezenet Network

Andrew Sebastian Lehman^{*1}, Andreas Widjaja^{2*}, Benny Budiawan Tjandrasa³, Joseph Sanjaya⁴

¹Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha

²Teknologi Informasi, Magister Ilmu Komputer, Universitas Kristen Maranatha

^{3,4}Bisnis Manajemen, Bisnis, Universitas Kristen Maranatha

Email: ¹andrewsebastianl@gmail.com, ²andreas.widjaja@it.maranatha.edu,

³benny.tjandrasa@gmail.com, ⁴sanjayajosep@gmail.com

(Naskah masuk: 19 Januari 2025, diterima untuk diterbitkan: 11 Juli 2025)

Abstrak: Lubang di jalan merupakan salah satu permasalahan utama yang dihadapi pengguna jalan, karena dapat menyebabkan kerusakan serius pada kendaraan serta meningkatkan risiko kecelakaan lalu lintas. Selain berdampak pada keselamatan pengemudi, terutama mereka yang tidak familiar dengan kondisi jalan, keberadaan lubang juga dapat memicu kemacetan dan meningkatkan emisi karbon akibat perlambatan lalu lintas. Untuk mengatasi permasalahan ini, studi ini mengusulkan sistem deteksi lubang secara real-time berbasis Convolutional Neural Network (CNN) dengan arsitektur ringan SqueezeNet. Sistem dirancang untuk bekerja dengan memanfaatkan gambar yang diambil oleh kamera kendaraan yang sedang bergerak, guna mengidentifikasi lubang yang berpotensi membahayakan. Model CNN dilatih menggunakan dataset gambar yang telah diberi label berdasarkan keberadaan lubang. Kinerja sistem dievaluasi menggunakan metrik precision, recall, dan F1-score guna mengukur keakuratannya dalam mendeteksi lubang. Hasil penelitian menunjukkan bahwa pendekatan ini berpotensi tinggi untuk diterapkan dalam sistem transportasi cerdas, guna meningkatkan keselamatan jalan serta mengurangi biaya perawatan kendaraan akibat kerusakan yang disebabkan oleh lubang jalan.

Kata Kunci – Komputer Visi; Lubang; CNN; SqueezeNet

Pothole Detection System Using Convolutional Neural Network with SqueezeNet Architecture

Abstract: Potholes are one of the major issues faced by road users, as they can cause serious damage to vehicles and increase the risk of traffic accidents. In addition to impacting driver safety – especially for those unfamiliar with road conditions – potholes can also lead to traffic congestion and increased carbon emissions due to traffic slowdowns. To address this problem, this study proposes a real-time pothole detection system based on a Convolutional Neural Network (CNN) using the lightweight SqueezeNet architecture. The system is designed to operate by utilizing images captured by cameras mounted on moving vehicles to identify potentially hazardous potholes. The CNN model is trained using a labeled image dataset indicating the presence or absence of potholes. The system's performance is evaluated using precision, recall, and F1-score metrics to assess its accuracy in detecting potholes. The research findings indicate that this approach has strong potential for implementation in intelligent transportation systems, aiming to enhance road safety and reduce vehicle maintenance costs caused by pothole-related damage.

Keywords – Computer Vision; Potholes; CNN; SqueezeNet

1. PENDAHULUAN

Lubang di jalan dapat menimbulkan konsekuensi serius, baik dari segi keselamatan maupun biaya. Biaya tersebut antara lain meningkatnya risiko kecelakaan, dampak negatif terhadap lingkungan, dan kemacetan lalu lintas. Selain itu, jalan berlubang dapat menyebabkan pemborosan waktu dan bahan bakar bagi pengemudi, dan dapat menjadi masalah yang membuat frustrasi investor di daerah dengan infrastruktur jalan yang buruk.

Lubang di jalan merupakan masalah keamanan utama dan membutuhkan deteksi dan perawatan yang cepat. Metode tradisional untuk mendeteksi lubang mengandalkan pemeriksaan manual yang subjektif, mahal, dan memakan waktu [1]. Dengan kemajuan dalam teknologi pemrosesan gambar digital, menggunakan komputer untuk memproses gambar melalui denoising, enhance, restore, segmentasi, dan fitur ekstraksi, seperti edge detection, threshold segmentation, dan morphological processing, telah banyak digunakan dalam deteksi lubang pada jalan ini. Tapi itu masih belum cukup akurat. Dengan munculnya deep learning, terutama jaringan Convolutional Neural Network, yang mampu mendeteksi objek dengan cepat dan akurat dengan mengekstraksi fitur secara otomatis, telah diterapkan di berbagai bidang termasuk infrastruktur transportasi, deteksi pejalan kaki, mengemudi tanpa awak, dan deteksi gangguan trotoar [2].

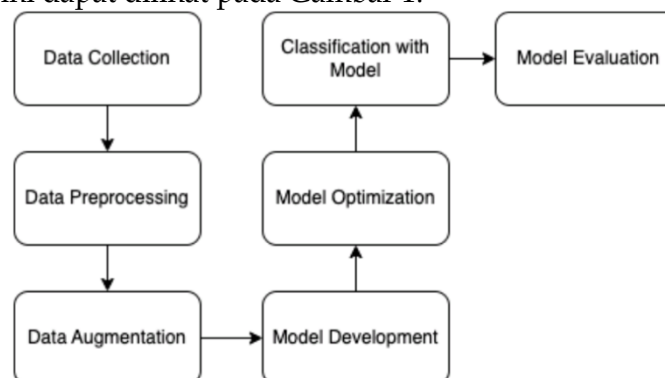
Berdasarkan kenyataan tersebut, pada penelitian ini akan dibuat satu model deep learning menggunakan metode CNN dengan jaringan SqueezeNet sehingga parameter tidak terlalu besar dan memakan waktu untuk deteksi [3]. Dataset yang digunakan berdasarkan 350 gambar tanpa lubang di jalan dan 330 gambar dengan lubang pada jalan yang akan dikumpulkan menggunakan metode manual. Tugas dari model tersebut adalah untuk mendeteksi lubang pada jalan.

2. METODE PENELITIAN

Tujuan dari penelitian yang diajukan adalah sebagai berikut :

- Implementasi dan menemukan konfigurasi yang efisien dan akurat untuk deteksi celah jalan menggunakan teknik CNN dengan arsitektur SqueezeNet.
- Mengevaluasi model yang telah di buat.

SqueezeNet adalah sebuah arsitektur jaringan saraf tiruan konvolusi (CNN) yang ringan dan didesain untuk tugas klasifikasi gambar realtime dan deteksi objek. Arsitektur ini didesain untuk memiliki jumlah parameter dan perhitungan yang lebih sedikit dibandingkan arsitektur CNN populer lainnya, tetapi masih mencapai akurasi tinggi. SqueezeNet dikembangkan dengan fokus pada pengurangan memori dan persyaratan komputasi dari jaringan, sehingga sesuai untuk implementasi pada perangkat dengan sumber daya komputasi terbatas. Jaringan ini mencapai pengurangan dengan menggunakan “modul api” yang melakukan ekstraksi fitur, dan “lapisan squeeze” yang mengurangi jumlah saluran dalam peta fitur, sehingga mengurangi memori dan jaringan. Teknologi ini belum pernah diterapkan untuk memecahkan masalah celah jalan menggunakan pengimajian termal. Rencana kami adalah untuk mengumpulkan data yang berisi termal jalan dengan dan tanpa celah jalan. Setelah ini berbagai teknik augmentasi data diterapkan untuk memperkaya data yang terkumpul untuk meningkatkan jumlah gambar dalam dataset. Akhirnya, klasifikasi menggunakan model jaringan saraf konvolusi (CNN) dilakukan dan hasil dibandingkan. Proses ini dapat dilihat pada Gambar 1.



Gambar 1. Research Methods

Proses pengumpulan data untuk gambar celah jalan melibatkan pengambilan set gambar yang komprehensif dan beragam dan secara akurat mewakili kondisi jalan dan celah jalan. Sangat penting untuk memastikan bahwa gambar memiliki kualitas tinggi dan diberi label dengan informasi yang relevan untuk memfasilitasi pelatihan dan evaluasi model deteksi celah jalan. Target pengumpulan data ini adalah 350 data gambar dengan anotasi celah jalan dan 330 gambar dengan anotasi tanpa

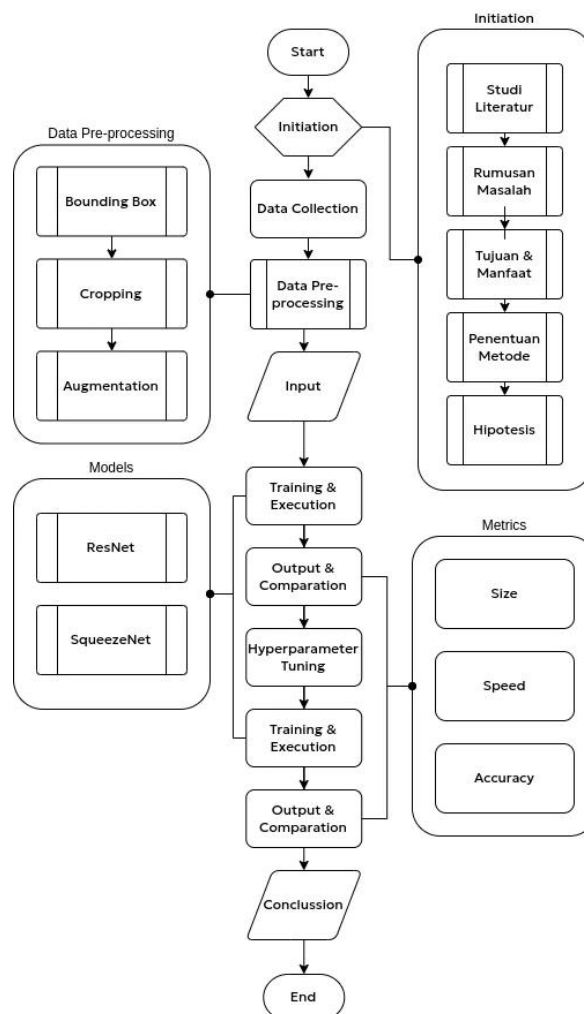
celah jalan. Untuk memastikan representatifitas data yang lebih besar dilakukan augmentasi data dengan menggunakan metode mix up dan cropping.

Pemanfaatan jalur udara dan laut belum bisa sepenuhnya menggantikan jalur distribusi darat yang menggunakan angkutan kendaraan roda empat dan lebih. Efisiensi energi bahan bakar minyak dan energi listrik sangat dipengaruhi oleh tingkat kemulusan jalan yang mayoritas menggunakan lapisan aspal. Tingkat friksi yang semakin tinggi pada lapisan aspal yang berlubang akan mengakibatkan tingkat efisiensi pemakaian bahan bakar minyak dan energi listrik semakin boros. Untuk itu demi menjaga tingkat efisiensi yang tinggi dalam pendistribusian baran, pemeliharaan lapisan aspal yang baik sangatlah diperlukan. Dari sudut pandang perekonomian guna mencegah kerusakan aspal menjadi semakin besar dan jalan hancur dimana diperlukan dana yang sangat besar dan waktu yang banyak untuk memperbaikinya, diperlukan suatu alat untuk mendeteksi kerusakan awal dari lapisan aspal di jalan raya sehingga perbaikan dapat segera dilaksanakan dengan biaya yang lebih murah dan waktu yang lebih singkat.

3. HASIL DAN PEMBAHASAN

3.1. Research Design

Penelitian ini menggunakan pendekatan kuantitatif untuk menguji dan mengevaluasi kinerja model berdasarkan tiga arsitektur yang memiliki kareakteristik ukuran, kecepatan, dan akurasi yang berbeda. Tahapan penelitian disajikan dalam Gambar 2.



Gambar 2. Research Design

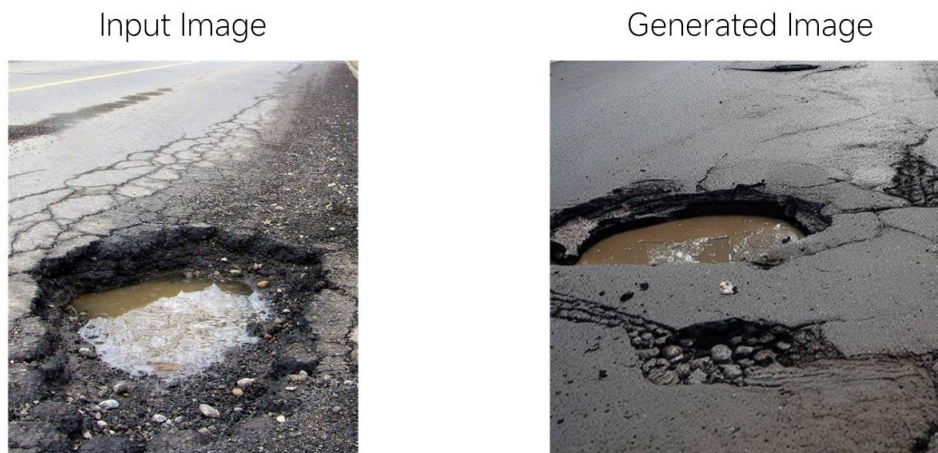
3.2. Preprocessing Data

Untuk mengembangkan jumlah data gambar yang digunakan, maka digunakan model generatif Star-GAN. Pemotongan gambar secara acak digunakan untuk meningkatkan ketahanan model terhadap perubahan posisi objek. Untuk memastikan kompatibilitas dengan model, ukuran gambar ditetapkan sebagai $224 \times 224 \times 3$ untuk semua model kecuali inceptionnet-v3, yang memerlukan ukuran input $299 \times 299 \times 3$ untuk lapisan konvolusi pertamanya.

Dalam upaya memaksimalkan efisiensi pelatihan dan pembelajaran, ukuran batch awal ditentukan sebagai 64 untuk semua model kecuali ResNet, yang dibatasi hingga 32 karena keterbatasan memori yang tersedia.

3.2.1. Star-Gan

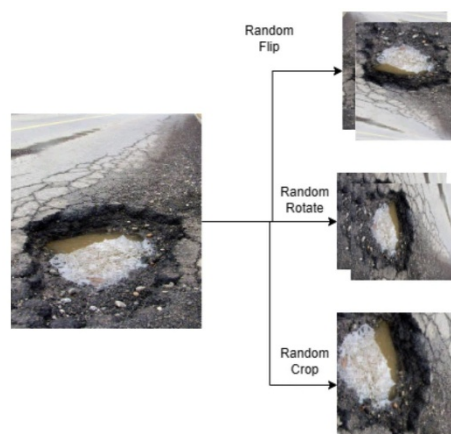
Metode augmentasi Star-Gan menciptakan gambar alami dengan mensintesis kelas yang serupa dalam dataset asli. Model Star-Gan dapat menghasilkan gambar yang ringan, cepat, dan alami dengan memanfaatkan satu model di beberapa domain. Metode augmentasi seperti Star-Gan dapat secara efektif memperbanyak dataset yang minimal untuk klasifikasi. Selain itu, menurut Hyun-Cheol Park pada penelitiannya menjelaskan bahwa metode ini dapat menciptakan lebih banyak gambar alami daripada Style-Gan, yang menghasilkan gambar secara buatan [28]. Hasil dari Star-Gan model dapat dilihat pada Gambar 3.



Gambar 3. Star-Gan Generated Image

3.2.2. Random Cropping & Rotate

Untuk memperkaya lebih lanjut dataset, dilakukan transformasi terhadap dataset gambar. Elemen transformasi yang terdiri dari flip vertikal, horizontal dan rotasi secara acak diterapkan pada sudut acak antara -90° hingga 90° untuk keragaman gambar. Pemotongan gambar secara acak digunakan untuk meningkatkan ketahanan model terhadap perubahan posisi objek Gambar 4.



Gambar 4. Random Flip, Rotate and Crop

Apabila kita perhatikan pada Gambar 4, penggunaan augmentasi data seperti random crop dan random rotation menghasilkan dataset baru yang dapat digunakan saat proses pelatihan.

Random rotation menghasilkan gambar yang dirotasi berdasarkan derajat yang dipilih secara acak. Setelah data yang telah di-augmentasi selesai digenerasi, dilakukan proses resize menggunakan salah satu fasilitas dari fast.ai. Metode resize yang digunakan adalah “squish,” dimana ukuran dimensi piksel gambar disesuaikan dengan frame yang telah ditentukan tanpa memperhatikan rasion aspek gambar sebelum dilakukan resize. Perbandingan hasil resize antara metode yang tersedia pada fast.ai, seperti padding, squish, dan crop, dapat dilihat pada Gambar 5.



Gambar 5. Perbedaan Resize Methods

Gambar 5 menunjukkan hasil resize 3 gambar dengan ukuran 15 x 15 piksel. Gambar pertama diresize menggunakan metode squish, gambar kedua dengan metode padding, dan gambar ketiga dengan metode crop.

Proses pembangunan arsitektur CNN bergantung pada input shape. Tanpa memiliki input shape yang tepat, arsitektur CNN tidak dapat dibentuk dengan baik. Oleh karena itu, dimensi gambar perlu disamakan sebelum dilakukan pelatihan.

3.2.3. Mixup

Metode mixup dapat meningkatkan regularitation model dalam computer vision. Dalam artikel jurnal mixup, Daojun Liang menawarkan metode training model dengan menggunakan campuran gambar dari set training. Misalnya, saat training model dengan dataset CIFAR-10, bukan gambar mentah yang dimasukkan ke model, melainkan dua gambar yang berbeda yang dikombinasikan secara linear menjadi satu tensor gambar baru, dengan formula:

$$newimage = t * image1 + (1 - t) * image2$$

Variabel t adalah antara 0 dan 1 yang merepresentasikan skala rasio antar ukuran gambar. Target label yang diberikan untuk gambar baru tersebut adalah campuran yang sama dari target asal dengan formula:

$$newtarget = t * target1 + (1 - t) * target2$$

Variabel t adalah antara 0 dan 1 yang merepresentasikan bobot pada label gambar. Proses ini menghasilkan gabungan linear antara dua gambar yang dapat membantu model belajar lebih efisien. Gambar 6 menunjukkan contoh input gambar yang dicampur dengan mixup.

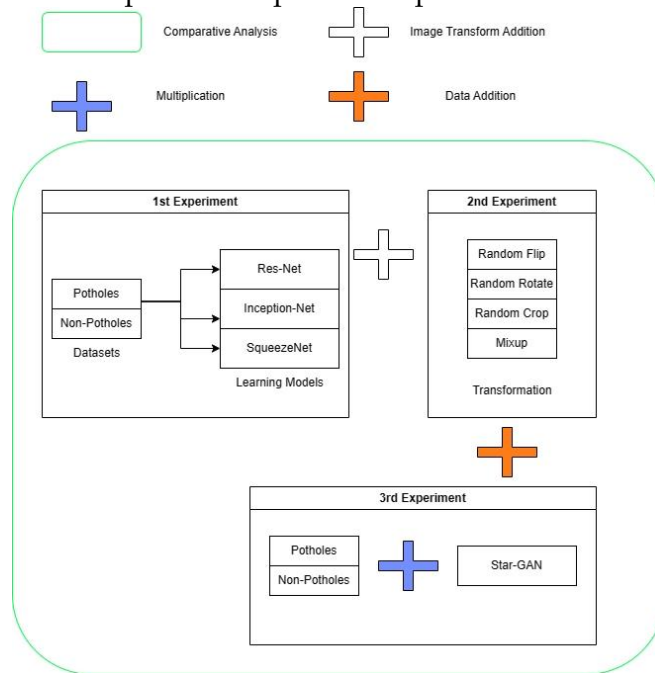


Gambar 6. Gambar Mixup

Gambar 6 menunjukkan bahwa gambar yang dicampur dengan mixup agak sulit untuk dipahami oleh mata manusia. Hal penting yang perlu diperhatikan adalah ketika training dengan mixup, final loss(training atau validasi) akan lebih besar daripada tanpa mixup, meskipun accuracy lebih baik. Hal ini karena model yang ditraining dengan mixup akan membuat prediksi yang lebih beragam dan robust [29].

3.3. Experiment Environment

Experiment dilakukan dengan menggunakan 3 model deep learning yang berbeda. PyTorch digunakan untuk proses pembelajaran model. Setiap model dijalankan dengan 200 epoch. Optimizer RMSprop dan adam optimizor dipilih karena memberikan kinerja yang optimal untuk masing-masing model. Skema eksperimen dapat dilihat pada Gambar 7.



Gambar 7. Skema Eksperimen

3.3.1. Evaluation Metrics

Metrik evaluasi yang digunakan untuk mengukur kinerja model adalah presisi, recall, F1-score, dan akurasi. Metrik-metrik ini didasarkan pada konsep true positif (TP), true negatif (TN), false positif (FP), dan false negatif (FN), yang merupakan hasil dari model klasifikasi biner.

- Presisi adalah proporsi prediksi positif yang benar. Presisi yang tinggi berarti model memiliki tingkat false positif yang rendah, yang berarti model tidak salah mengklasifikasikan instansi negatif sebagai positif. Rumus untuk presisi adalah :

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Recall adalah proporsi instansi positif yang diprediksi dengan benar. Recall yang tinggi berarti model memiliki tingkat false negatif yang rendah, yang berarti model tidak melewatkan instansi positif. Rumus untuk recall adalah:

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

- F1-score adalah rata-rata harmoni dari presisi dan recall. F1-score yang tinggi berarti model memiliki presisi dan recall yang tinggi, yang berarti model akurat dan konsisten dalam prediksinya. Rumus untuk F1-score adalah:

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

Akurasi adalah proporsi yang benar. Akurasi yang tinggi berarti model memiliki tingkat kesalahan yang rendah, yang berarti model dapat mengklasifikasikan baik instansi positif maupun negatif dengan benar. Rumus untuk akurasi adalah :

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

- d. Model size adalah ukuran model setelah dilakukannya pelatihan, hal ini penting karena model akhir akan digunakan pada device yang bisa digunakan secara mobile.

Metrik-metrik ini berguna untuk mengevaluasi kinerja model dari berbagai aspek

3.4. Hyperparameter Tuning (Batch Size)

Batch size adalah hyper-parameter yang penting training model. Batch size menentukan jumlah data yang diproses dalam satu iterasi. Leslie menyarankan untuk menggunakan batch size sebesar mungkin agar bisa menggunakan learning rate yang lebih tinggi [30]. Biasanya, batch size besar lebih disukai karena bisa menggunakan learning rate yang lebih tinggi [30].

Untuk mengetahui batch size terbesar yang bisa digunakan, bisa dilakukan eksperimen dengan mengurangi batch size secara bertahap sampai tidak ada error OOM. Error OOM terjadi ketika memori hardware tidak cukup untuk menampung data. Eksperimen ini dilakukan dengan batch size 256 dan dikurangi 32 setiap kali sampai tidak ada error OOM. Hasil eksperimen ini dilakukan terhadap semua learning models yang telah ditentukan pada 3.3 bisa dilihat pada Tabel 1.

Tabel 1. Hyperparameter Tuning (Batch Size)

Batch Size	ResNet	InceptionNet	SqueezeNet
256	OOM	OOM	OOM
224	OOM	OOM	OOM
192	OOM	OOM	OOM
160	OOM	OOM	OOM
128	OOM	OOM	OOM
96	OOM	OOM	OOM
64	OOM	Success	Success
32	Success	Success	Success

3.5. Experiment Result

Tiga percobaan dilakukan. Pada percobaan pertama tiga model berbasis deep learning digunakan tanpa menerapkan transformasi dan augmentasi. Pada percobaan kedua, hanya transformasi yang diterapkan. Pada percobaan ketiga dataset diperkaya dengan menggunakan Star-Gan.

3.5.1. Hasil Tanpa Transformasi dan Augmentasi

Pada sub bab ini, dilakukan eksperimen untuk melatih learning model yang dilakukan. Eksperimen ini dilakukan tanpa aplikasi transformasi atau augmentasi. Hal ini dilakukan untuk melihat perbandingan performa antara kedua metode sehingga didapatkan metode training yang memiliki performa terbaik dalam metrik yang telah ditentukan sebelumnya. Hasil eksperimen ini dapat dilihat pada Tabel 2.

Tabel 2 membandingkan kinerja tiga model berdasarkan penggunaan tanpa transformasi dan augmentasi berbasis GAN. Selain itu, tabel ini juga menunjukkan perbedaan kinerja antara model-model tersebut dalam data gambar. Dari Tabel 2, dapat dilihat bahwa rata-rata akurasi terendah adalah 61,11% untuk ketiga model tanpa transformasi. Model dengan akurasi tertinggi adalah InceptionNet 86,40%. Berikut juga hasil yang didapatkan terkait dengan waktu yang dibutuhkan untuk pelatihan dan model size.

Tabel 2. Hasil Tanpa Transformasi dan Augmentasi

Learning Model	Epoch	Train Loss	Valid Loss	Precision	Recall	F1	Acc	Avg. Acc
ResNet	1	5.3364	4.3536	0.184	0.172	0.182	0.185	0.493
	40	4.1546	3.2342	0.254	0.265	0.283	0.285	
	80	3.2345	2.8534	0.425	0.471	0.431	0.495	
	120	2.2151	2.1324	0.587	0.579	0.581	0.513	
	160	0.9412	1.0352	0.631	0.645	0.662	0.689	
	200	0.5432	0.7435	0.784	0.772	0.802	0.791	
InceptionNet	1	4.9364	4.2145	0.206	0.214	0.209	0.215	0.573
	40	4.0341	3.1533	0.314	0.325	0.381	0.385	
	80	2.9156	2.3215	0.495	0.511	0.536	0.491	
	120	1.8155	1.4515	0.682	0.671	0.681	0.673	
	160	0.7352	1.0452	0.811	0.801	0.816	0.809	
	200	0.4132	0.3432	0.884	0.871	0.899	0.864	
SqueezeNet	1	6.1254	4.4137	0.121	0.154	0.151	0.155	0,438
	40	4.0623	3.3512	0.254	0.265	0.283	0.285	
	80	3.1341	2.9538	0.425	0.471	0.431	0.495	
	120	2.0154	2.0321	0.487	0.479	0.581	0.513	
	160	0.8232	0.7352	0.561	0.552	0.562	0.569	
	200	0.7532	0.6965	0.694	0.671	0.601	0.611	

Tabel 3. Evaluasi Summary Tanpa Transformasi dan Augmentasi

Learning Model	Accuracy	Model Size (MB)	Speed (Minutes)
ResNet	0.791	184.81	26.1
InceptionNet	0.864	96.32	29.5
SqueezeNet	0.611	19.84	17.4

3.5.2. Hasil dengan Transformasi

Pada subbab ini dilakukan eksperimen untuk melatih learning model yang ditentukan. Eksperimen dilakukan tanpa aplikasi transformasi atau augmentasi. Hal ini dilakukan untuk melihat perbandingan performa antara kedua metode sehingga didapatkan metode training yang memiliki performa terbaik dalam metrics yang telah ditentukan sebelumnya. Hasil dari eksperimen diapaparkan pada tabel 4.

Tabel 4. Hasil dengan Tranformasi

Learning Model	Epoch	Train Loss	Valid Loss	Precision	Recall	F1	Acc	Avg. Acc
ResNet	1	5.1315	4.2241	0.199	0.185	0.187	0.190	0.522
	40	3.8315	3.8155	0.275	0.291	0.289	0.282	
	80	3.0125	2.7518	0.491	0.492	0.487	0.485	
	120	2.0155	1.9125	0.698	0.672	0.682	0.611	
	160	0.8114	0.8323	0.743	0.735	0.781	0.761	
	200	0.4653	0.4432	0.804	0.814	0.822	0.803	
InceptionNet	1	4.9125	4.2061	0.214	0.216	0.225	0.221	0.603
	40	3.7151	2.8155	0.314	0.325	0.381	0.385	
	80	2.5432	2.3124	0.512	0.521	0.526	0.533	
	120	1.5321	1.3215	0.711	0.732	0.743	0.723	
	160	0.6231	0.9523	0.843	0.832	0.836	0.849	
	200	0.3732	0.3132	0.904	0.902	0.891	0.904	
SqueezeNet	1	5.1341	4.3124	0.152	0.163	0.166	0.167	0,506
	40	3.7564	3.3252	0.310	0.271	0.291	0.312	
	80	3.0151	2.8125	0.495	0.490	0.501	0.511	
	120	1.8141	1.9125	0.612	0.631	0.642	0.636	
	160	0.6432	0.6213	0.701	0.682	0.705	0.700	
	200	0.5515	0.5923	0.715	0.701	0.703	0.711	

Tabel 4 menunjukkan hasil eksperimen dengan menggunakan transformasi tanpa augmentasi. Tabel ini membandingkan kinerja tiga model berdasarkan penggunaan dengan transformasi, dan juga tabel ini menunjukkan perbedaan kinerja antara model-model tersebut dalam data gambar. Dapat dilihat bahwa rata-rata akurasi terendah adalah 71,11% untuk ketiga model tanpa transformasi. Model dengan akurasi tertinggi adalah InceptionNet (90,04%). Berikut hasil terkait waktu yang dibutuhkan untuk pelatihan dan model size.

Tabel 5. Evaluasi Summary dengan Transformasi

Learning Model	Accuracy	Model Size	Speed (Minutes)
ResNet	0.803	185.12	27.5
InceptionNet	0.904	97.55	30.6
SqueezeNet	0.711	19.91	17.1

3.5.3. Hasil dengan Transformasi dan Augmentasi

Pada subbab ini dilakukan eksperimen untuk melatih learning model yang ditentukan. Eksperimen dilakukan menggunakan aplikasi transformasi dan augmentasi. Hal ini dilakukan untuk melihat perbandingan antara performa antara kedua metode sehingga didapatkan metode training yang memiliki performa terbaik dalam metrics yang telah ditentukan sebelumnya. Hasil dapat dilihat pada tabel 6.

Tabel 6. Hasil dengan Transformasi dan Augmentasi

Learning Model	Epoch	Train Loss	Valid I Loss	Precision	Recall	F1	Acc	Avg. Acc
ResNet	1	4.9151	4.1251	0.191	0.196	0.195	0.193	0.57
	40	3.6136	3.6314	0.301	0.312	0.315	0.321	
	80	2.9104	2.5235	0.531	0.512	0.514	0.531	
	120	1.8151	1.7124	0.715	0.723	0.731	0.721	
	160	0.6123	0.5124	0.792	0.801	0.813	0.801	
	200	0.4012	0.4125	0.846	0.855	0.841	0.853	
InceptionNet	1	4.6431	4.1515	0.221	0.215	0.215	0.233	0.64
	40	3.5325	2.7151	0.351	0.371	0.385	0.391	
	80	2.4125	2.2151	0.551	0.574	0.571	0.583	
	120	1.2241	1.3151	0.802	0.781	0.795	0.803	
	160	0.5234	0.6234	0.863	0.871	0.881	0.894	
	200	0.2915	0.2781	0.915	0.931	0.901	0.933	
SqueezeNet	1	5.0165	4.2145	0.156	0.151	0.167	0.171	0.561
	40	3.5123	3.1616	0.361	0.315	0.371	0.377	
	80	2.5132	2.6135	0.525	0.512	0.589	0.581	
	120	1.6561	1.7161	0.651	0.679	0.688	0.686	
	160	0.5781	0.5918	0.755	0.731	0.741	0.761	
	200	0.4516	0.4156	0.776	0.781	0.791	0.793	

Tabel 6 membandingkan kinerja tiga model berdasarkan penggunaan dengan transformasi dan augmentasi. Selain itu tabel ini juga menunjukkan perbedaan kinerja antara mode-model tersebut dalam data gambar. Dari tabel 6 dapat dilihat bahwa rata-rata akurasi terendah adalah 79,30% untuk ketiga model tanpa transformasi. Model dengan akurasi tertinggi adalah INceptionNet (93,30%). Berikut hasil yang didapatkan terkait waktu yang dibutuhkan untuk pelatihan dan model size.

Tabel 7. Evaluasi Summary dengan Transformasi dan Augmentasi

Learning Model	Accuracy	Model Size	Speed (Minutes)
ResNet	0.853	204.13	27.5
InceptionNet	0.933	101.5	30.1
SqueezeNet	0.793	21.14	18.1

3.6. Rekapitulasi Hasil Penelitian

Pada bagian ini dilakukan analisis akhir dari model yang diuji dengan berbagai macam eksperimen. Semua eksperimen yang dilakukan menggunakan perangkat keras yang memiliki kemampuan yang sama. Berdasarkan hasil eksperimen pada Bab 3.5 didapatkan data rekapitulasi yang dipaparkan pada Tabel 7.

- Tabel Rekapitulasi Hasil Penelitian menunjukkan nilai metrik akurasi, ukuran model, dan kecepatan untuk tiga model yang digunakan (ResNet, I dan InceptionNet).
- Pada analisis Model CNN telah dianalisis kinerja dan karakteristik dari masing-masing model CNN yang digunakan. Ditemukan bahwa SqueezeNet memiliki ukuran model dan kecepatan terbaik dengan akurasi rendah. Dapat disimpulkan bahwa penggunaan arsitektur SqueezeNet terhadap model CNN sangat cocok untuk penggunaan realtime untuk tugas yang tidak terlalu membutuhkan akurasi. Keunggulan SqueezeNet pada metrik ukuran model dan kecepatan menunjukkan bahwa model ini sangat cocok untuk implementasi pada bidah IoT (Internet of Things). Diketahui juga bahwa InceptionNet memiliki akurasi tertinggi serta ukuran model dan kecepatan yang lebih baik dari ResNet. Dapat disimpulkan bahwa model CNN dengan arsitektur InceptionNet merupakan model CNN yang memiliki performa terbaik diantara ketiga arsitektur yang diuji.
- Dalam bagian Analisis Data Augmentation dianalisis pengaruh dari data augmentation yang dilakukan dengan menggunakan Star-GAN. Ditemukan bahwa data augmentation dengan Star-GAN memberikan peningkatan yang lebih baik pada kinerja model, terutama pada InceptionNet. Dapat disimpulkan bahwa data augmentation dengan Star-GAN sangat membantu model ketika memiliki limitasi jumlah data karena menghasilkan gambar yang lebih natural dan variatif.

4. KESIMPULAN

Penelitian ini menghasilkan kesimpulan bahwa: Model CNN yang menggunakan ResNet sebagai dasar arsitekturnya mampu menghasilkan akurasi yang cukup memuaskan dengan mengabaikan model size dan kecepatan. Tabel rekapitulasi menunjukkan bukti bahwa model CNN dengan arsitektur ResNet memiliki akurasi (85.3%), yang merupakan nilai yang baik, tetapi memiliki model size (204.13 MB) dan kecepatan (27.5 Menit), yang merupakan nilai yang buruk, jika dibandingkan dengan model-model lain yang diuji dalam penelitian ini.

Hipotesis kedua yang menyatakan bahwa model CNN yang dibangun dengan menggunakan SqueezeNet dapat memiliki model size dan kecepatan yang paling optimal dengan mengorbankan akurasi terbukti valid. Hal ini dapat dilihat pada tabel rekapitulasi yang menampilkan bahwa model CNN dengan arsitektur SqueezeNet memiliki model size (19.91 MB) dan kecepatan (18.1 Menit) yang paling baik diantara tiga model lainnya, tetapi memiliki akurasi (79.3%) yang paling rendah.

Model CNN telah berhasil diimplementasikan dan dievaluasi untuk tugas pengenalan lubang pada jalan dengan menggunakan TTA. Hasilnya, ResNet 152 memiliki accuracy (85.3%), InceptionNet memiliki akurasi yang paling baik (93.3%), dan SqueezeNet memiliki akurasi yang paling buruk (79.3%). Model CNN dengan arsitektur EfficientNet menunjukkan kinerja yang paling baik pada metrics akurasi, model size, dan kecepatan dibandingkan dengan dua model lainnya. SqueezeNet memiliki model size dan kecepatan yang paling rendah sehingga sangat sesuai untuk digunakan dalam implementasi realtime dengan mengabaikan accuracy.

Penelitian ini tidak luput dari beberapa kekurangan yang harus dijadikan bahan pertimbangan untuk penelitian mendatang, yaitu: Model CNN dengan arsitektur SqueezeNet belum mencapai performa yang optimal dan membutuhkan optimisasi yang lebih mendalam agar dapat memperbaiki akurasi-nya. Model CNN yang sudah dibuat hendaknya diuji coba pada

aplikasi yang sudah siap beroperasi dalam lingkungan nyata sehingga dapat dianalisis kinerjanya terhadap kasus-kasus yang sesuai dengan tujuan penelitian.

DAFTAR PUSTAKA

- [1] S. K. Sharma and R. C. Sharma, "Pothole Detection and Warning System for Indian Roads," in *Advances in Interdisciplinary Engineering*, M. Kumar, R. K. Pandey, and V. Kumar Eds. Singapore: Springer Singapore, 2019, pp. 511-519.
- [2] M. P. B and S. K.C, "Enhanced pothole detection system using YOLOX algorithm," (in en), *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 22, 2022-08-31 2022, doi: 10.1007/s43684-022-00037-z.
- [3] C. Li, X. Wei, H. Yu, J. Guo, X. Tang, and Y. Zhang, "An Enhanced SqueezeNet Based Network for Real-Time Road-Object Segmentation," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 12/2019 2019, Xiamen, China: IEEE, pp. 1214-1218, doi: 10.1109/SSCI44817.2019.9002818.
- [4] A. A. Alhussan, D. S. Khafaga, E.-S. M. El-Kenawy, A. Ibrahim, M. M. Eid, and A. A. Abdelhamid, "Pothole and Plain Road Classification Using Adaptive Mutation Dipper Throated Optimization and Transfer Learning for Self Driving Cars," *IEEE Access*, vol. 10, pp. 84188-84211, 2022 2022, doi: 10.1109/ACCESS.2022.3196660.
- [5] A. K. Pandey, R. Iqbal, T. Maniak, C. Karyotis, S. Akuma, and V. Palade, "Convolution neural networks for pothole detection of critical road infrastructure," (in en), *Computers and Electrical Engineering*, vol. 99, p. 107725, 04/2022 2022, doi: 10.1016/j.compeleceng.2022.107725.
- [6] E. Moscoso Thompson et al., "SHREC 2022: Pothole and crack detection in the road pavement using images and RGB-D data," (in en), *Computers & Graphics*, vol. 107, pp. 161-171, 10/2022 2022, doi: 10.1016/j.cag.2022.07.018.
- [7] Aparna, Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks based potholes detection using thermal imaging," (in en), *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 578-588, 03/2022 2022, doi: 10.1016/j.jksuci.2019.02.004.
- [8] A. Yang Her, W. Kean Yew, P. Jia Yew, and M. Chong Jia Ying, "Real-time pothole detection system on vehicle using improved YOLOv5 in Malaysia," in *IECON 2022 - 48th Annual Conference of the IEEE Industrial Electronics Society*, 2022-10-17 2022, Brussels, Belgium: IEEE, pp. 1-5, doi: 10.1109/IECON49645.2022.9968423.
- [9] U. Yahya, M. Lucky, M. Mansoor, N. Sharifah, A. Kasule, and K. Usama, "IoT-Based Pothole Mapping Agent with Remote Visualization," 2022 2022, doi: 10.48550/ARXIV.2212.14764
- [10] A. K. Pandey, R. Iqbal, S. Amin, T. Maniak, V. Palade, and C. Karyotis, "Deep Neural Networks Based Approach for Pothole Detection," in *2021 4th International Conference on Signal Processing and Information Security (ICSPIS)*, 2021-11-24 2021, Dubai, United Arab Emirates: IEEE, pp. 1-4, doi: 10.1109/ICSPIS53734.2021.9652434.
- [11] K. R. Ahmed, "Smart Pothole Detection Using Deep Learning Based on Dilated Convolution," (in en), *Sensors*, vol. 21, no. 24, p. 8406, 2021-12-16 2021, doi: 10.3390/s21248406.
- [12] A. A. Shaghouri, R. Alkhatib, and S. Berjaoui, "Real-Time Pothole Detection Using Deep Learning," 2021-07-13 2021.
- [13] S. Kharel and K. R. Ahmed, "Potholes Detection Using Deep Learning and Area Estimation Using Image Processing," in *Intelligent Systems and Applications*, vol. 296, K. Arai Ed. Cham: Springer International Publishing, 2022, pp. 373-388.
- [14] E. S. Team. "What is Machine Learning? A definition." <https://expertsystem.com/machine-learning-definition> (accessed Jun. 8, 2020).
- [15] A. Burkov, *The hundred-page machine learning book*. Andriy Burkov (in English), 2019.

- [16] F. Ciabuschi and B. Venkateswaran, *Neural networks with R: smart models using CNN, RNN, deep learning, and artificial intelligence principles*. Birmingham, UK Mumbai: Packt (in eng), 2017, p. 256.
- [17] P. Srivastava and A. Rai. "Supervised and Unsupervised learning." <https://www.geeksforgeeks.org/supervised-unsupervised-learning/> (accessed Jun. 08, 2020).
- [18] J. Andreanus and A. Kurniawan, "Sejarah, Teori Dasar dan Penerapan Reinforcement Learning: Sebuah Tinjauan Pustaka," *Jurnal Telematika*, vol. 12, no. 2, pp. 113-118, 2018
- [19] V. Dhar, "Data science and prediction," (in en), *Communications of the ACM*, vol. 56, no. 12, pp. 64-73, 12/2013 2013, doi: 10.1145/2500499.
- [20] H. J. Adèr, G. J. Mellenbergh, and D. J. Hand, *Advising on research methods: a consultant's companion*. Huizen: Johannes van Kessel (in eng), 2008, p. 574.
- [21] A. W. Trask, *Grokking deep learning*. Shelter Island: Manning, 2019, p. 309.
- [22] J. D. Kelleher, *Deep learning (The MIT press essential knowledge series)*. Cambridge, Massachusetts: The MIT Press, 2019, p. 280.
- [23] F. Malik, "Understanding Neural Networks: From Activation Function To Back Propagation," in *Medium*, ed, 2018.
- [24] G. Bonaccorso, *Mastering machine learning algorithms: expert techniques to implement popular machine learning algorithms and fine-tune your models*. (in English), 2018.
- [25] R. Venkatesan and B. Li, *Convolutional neural networks in visual computing: a concise guide (Data-enabled engineering)*. Boca Raton: CRC Press, Taylor & Francis Group, CRC Press is an imprint of the Taylor & Francis Group, an informa business, 2018, p. 168.
- [26] A. Karpathy, "Cs231n convolutional neural networks for visual recognition," *Neural networks*, vol. 1, no. 1, 2016 2016.
- [27] S. Qiu, X. Xu, and B. Cai, "FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks," 2017 2017, doi: 10.48550/ARXIV.1706.08098.
- [28] H.-C. Park, I.-P. Hong, S. Poudel, and C. Choi, "Data Augmentation Based on Generative Adversarial Networks for Endoscopic Image Classification," *IEEE Access*, vol. 11, pp. 49216-49225, 2023 2023, doi: 10.1109/ACCESS.2023.3275173.
- [29] D. Liang, F. Yang, T. Zhang, and P. Yang, "Understanding Mixup Training Methods," *IEEE Access*, vol. 6, pp. 58774-58783, 2018 2018, doi: 10.1109/ACCESS.2018.2872698
- [30] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay," 2018 2018, doi: 10.48550/ARXIV.1803.09820.