

Alat Pendeteksi Kerusakan Jalan Berlubang Berbasis Convolutional Neural Network

Adzkia Faradiba Eyila Putri

Teknik Mekatronik, Surabaya, Indonesia

Abstrak

Penelitian ini mengembangkan sistem pendeteksi kerusakan lubang jalan berbasis CNN dengan menggunakan arsitektur YOLO, dengan tujuan untuk dapat menghitung jumlah lubang pada gambar yang diambil pada jalan yang dilalui. Pengembangan sistem melibatkan integrasi antara penggunaan web-cam dan algoritma CNN pada aplikasi Visual Studio Code. Sistem deteksi ini menggunakan model YOLO (You Only Look Once) untuk memproses gambar beresolusi tinggi yang diambil selama perjalanan. Proses pengumpulan data dilakukan dengan mengambil gambar lubang jalan yang kemudian diberi label dan dibagi menjadi data latih dan uji. Hasil pelatihan model dievaluasi menggunakan data validasi dan pengujian untuk memastikan akurasi deteksi.

Sejarah Artikel

Submitted: 25 Juni 2024

Accepted: 30 Juni 2024

Published: 1 July 2024

Kata Kunci

Deteksi lubang berbasis CNN, Arsitektur YOLO, sistem deteksi kerusakan lubang

Penghitungan lubang jalan, Pemrosesan gambar resolusi tinggi, Integrasi kamera web, aplikasi Visual Studio Code, Pelabelan gambar.

Pendahuluan

Lubang di jalan menimbulkan bahaya besar bagi kendaraan dan pengemudi, sehingga menyebabkan potensi kecelakaan dan meningkatkan biaya pemeliharaan. Metode tradisional untuk mendeteksi dan memperbaiki lubang sering kali memakan banyak tenaga dan tidak efisien. Terdapat kebutuhan akan sistem yang otomatis, akurat, dan efisien untuk mendeteksi dan mengukur kerusakan lubang secara real-time. Penelitian ini menjawab kebutuhan tersebut dengan mengembangkan sistem pendeteksi jalan berlubang berbasis CNN menggunakan arsitektur YOLO, yang mengintegrasikan web-cam dan aplikasi Visual Studio Code untuk memproses gambar resolusi tinggi yang diambil selama perjalanan. Sistem ini bertujuan untuk memberikan deteksi dan penghitungan lubang yang akurat untuk memfasilitasi pemeliharaan jalan yang tepat waktu dan meningkatkan keselamatan jalan (Redmon et al., 2016; He et al., 2016)

Mengeksplorasi keefektifan sistem pendeteksi jalan berlubang berbasis CNN menggunakan arsitektur YOLO dalam mendeteksi dan menghitung jalan berlubang secara akurat pada gambar resolusi tinggi yang diambil selama perjalanan, beserta dampak pengintegrasian web-cam dan aplikasi Visual Studio Code terhadap kinerja sistem. , mengungkapkan kemajuan signifikan dalam pemeliharaan dan keselamatan jalan secara real-time. Penelitian ini menyelidiki bagaimana pendekatan inovatif ini dapat merevolusi deteksi lubang jalan, memberikan wawasan mengenai keakuratan dan efisiensi sistem,

dan mengevaluasi manfaat integrasi teknologi yang lancar untuk meningkatkan pengelolaan infrastruktur jalan.

Tinjauan Literatur

Deteksi lubang jalan telah menjadi bidang penelitian yang penting karena dampaknya yang signifikan terhadap keselamatan jalan dan biaya pemeliharaan. Metode tradisional, yang sering kali melibatkan inspeksi manual, memakan waktu, padat karya, dan rentan terhadap kesalahan manusia. Kemajuan terkini dalam visi komputer dan pembelajaran mendalam telah membuka jalan bagi sistem deteksi lubang otomatis, menawarkan alternatif yang lebih efisien dan akurat.

Pembelajaran Mendalam dan CNN dalam Pemrosesan Gambar

Jaringan Neural Konvolusional (CNN) telah merevolusi tugas pemrosesan gambar karena kemampuannya mempelajari dan mengekstrak fitur yang relevan dari gambar secara otomatis. CNN telah berhasil diterapkan pada berbagai aplikasi seperti deteksi objek, klasifikasi gambar, dan segmentasi. Krizhevsky dkk. (2012) mendemonstrasikan efektivitas CNN dalam klasifikasi gambar dengan diperkenalkannya AlexNet, yang mencapai akurasi yang belum pernah terjadi sebelumnya pada kumpulan data ImageNet. Berdasarkan hal ini, pengembangan arsitektur yang lebih canggih seperti VGGNet (Simonyan &

Zisserman, 2014) dan ResNet (He et al., 2016) semakin meningkatkan kinerja CNN dalam berbagai tugas visual.

Arsitektur YOLO untuk Deteksi Objek Real-Time

You Only Look Once (YOLO) adalah sistem pendeteksi objek canggih yang dikembangkan oleh Redmon dkk. (2016). Tidak seperti metode deteksi objek tradisional yang menggunakan jaringan proposal wilayah untuk menghasilkan kotak pembatas potensial, YOLO mengubah deteksi objek menjadi masalah regresi tunggal, yang secara langsung memprediksi kotak pembatas dan probabilitas kelas dari gambar penuh dalam satu evaluasi. Pendekatan ini secara signifikan meningkatkan kecepatan deteksi, menjadikan YOLO cocok untuk aplikasi real-time.

Deteksi Lubang Menggunakan CNN dan YOLO

Beberapa penelitian telah mengeksplorasi penerapan CNN dan YOLO dalam deteksi lubang. Zhang dkk. (2018) mengusulkan pendekatan pembelajaran mendalam menggunakan R-CNN Lebih Cepat untuk deteksi lubang, mencapai akurasi tinggi tetapi dengan mengorbankan kecepatan pemrosesan yang lebih lambat. Untuk mengatasi masalah kecepatan, Liu dkk. (2019) memanfaatkan arsitektur YOLO untuk deteksi lubang secara real-time, menunjukkan bahwa efisiensi dan akurasi YOLO menjadikannya pilihan ideal untuk tugas ini.

Integrasi Web-Cam dan Visual Studio Code dalam Sistem Deteksi Lubang

Mengintegrasikan kamera web dengan sistem deteksi berbasis CNN memungkinkan pengambilan gambar beresolusi tinggi secara terus-menerus, yang sangat penting untuk deteksi lubang berlubang secara akurat. Penggunaan aplikasi seperti Visual Studio Code memfasilitasi pemantauan jarak jauh dan pengumpulan data, memungkinkan integrasi yang lancar dan pemrosesan waktu nyata. Kombinasi ini memastikan bahwa gambar diambil dan dianalisis dengan cepat, sehingga meningkatkan efisiensi sistem secara keseluruhan.

Pengumpulan Data dan Evaluasi Model

Pengumpulan data dan evaluasi model yang efektif merupakan komponen penting dari sistem pembelajaran mesin apa pun. Prosesnya melibatkan pengambilan gambar permukaan jalan, memberi label pada lubang, dan membagi kumpulan data menjadi kumpulan pelatihan dan pengujian. Performa model yang dilatih kemudian dievaluasi menggunakan data validasi dan pengujian untuk

memastikan keakuratan dan keandalannya. Teknik seperti validasi silang, presisi, perolehan kembali, dan skor F1 adalah metrik yang umum digunakan untuk menilai kinerja model (Sokolova & Lapalme, 2009).

Integrasi CNN dan YOLO untuk deteksi lubang memberikan solusi yang menjanjikan terhadap tantangan yang terkait dengan metode tradisional. Dengan memanfaatkan gambar resolusi tinggi yang diambil melalui kamera web dan memanfaatkan aplikasi seperti Visual Studio Code, penelitian ini bertujuan untuk mengembangkan sistem yang kuat dan efisien untuk pendeteksian lubang secara real-time. Kemajuan berkelanjutan dalam pembelajaran mendalam dan teknik visi komputer diharapkan dapat lebih meningkatkan akurasi dan efektivitas sistem tersebut, yang pada akhirnya berkontribusi pada peningkatan praktik keselamatan dan pemeliharaan jalan raya.

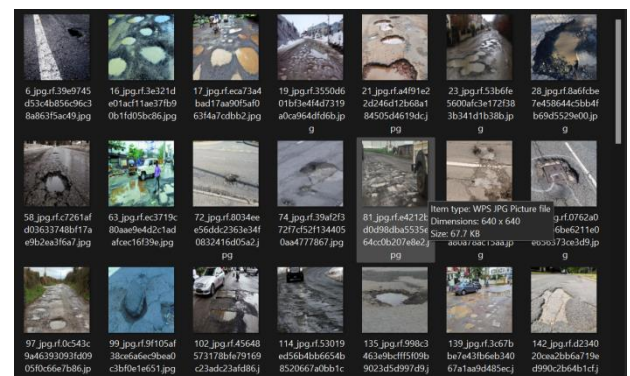
Metode

Penelitian ini melibatkan pengembangan dan implementasi sistem deteksi kerusakan lubang jalan berbasis CNN dengan menggunakan arsitektur YOLO (You Only Look Once). Sistem ini bertujuan untuk mendeteksi dan menghitung secara akurat jumlah lubang pada gambar resolusi tinggi yang diambil selama perjalanan. Metodologi ini dibagi menjadi beberapa tahapan utama: pengumpulan data, pengembangan sistem, pelatihan model, dan evaluasi.

Pengumpulan data

Gambar 1

kumpulan data kerusakan lubang

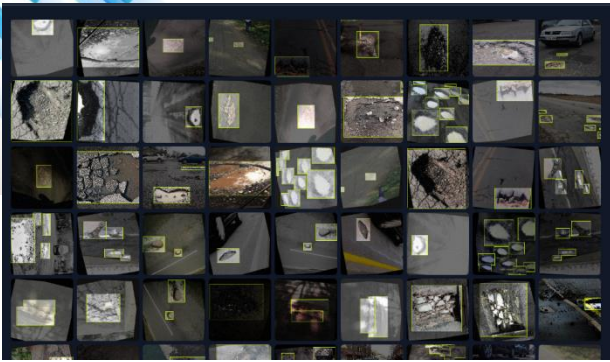


Data set yang digunakan merupakan dataset yang diperoleh dari beberapa referensi, seperti dataset Kaggle dan juga Roboflow

Pelabelan:

Angka2

memberi label pada gambar



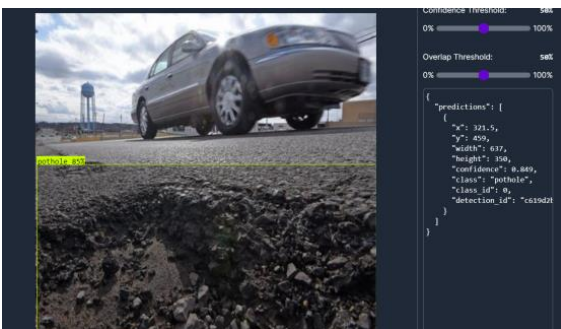
Gambar yang diambil diberi label secara manual untuk mengidentifikasi lokasi dan ukuran lubang. Proses pelabelan ini melibatkan pembuatan kotak pembatas di sekitar lubang dan pemberian label kelas yang sesuai. Data yang diberi label kemudian dibagi menjadi dua set: data pelatihan (80%) dan data pengujian (20%).

Anotasi kumpulan data:

Setelah mengatur ulang ukuran dan format gambar agar seragam, kumpulan data diberi anotasi untuk memberikan label yang mengidentifikasi karakteristik klasifikasi objek yang akan dideteksi. Dalam hal ini objek tersebut tergolong jalan yang memenuhi kriteria berlubang. Proses anotasi dataset ini dapat dilakukan dengan menggunakan aplikasi Roboflow.

Angka3

Anotasi Kumpulan Data Dengan Pelabelan



Pengembangan sistem

Integrasi Model YOLO:

Angka4

Pengujian Model

```

1 from inference.models.utils import get_robotflow_model
2 import cv2
3
4 # Roboflow model
5 model_name = "jalan_rusak"
6 model_version = "1"
7
8 # Open the default camera (usually the built-in webcam)
9 cap = cv2.VideoCapture(0)
10
11 # Check if the webcam is opened successfully
12 if not cap.isOpened():
13     print("Error: Could not open camera.")
14     exit()
15
16 # Get Roboflow face model (this will fetch the model from Roboflow)
17 model = get_robotflow_model(
18     model_id="{}/{}".format(model_name, model_version),
19     #Replace ROBOFLOW_API_KEY with your Roboflow API Key
20     api_key="42lnf93ktwwPtOoLGGKN"
21 )
22
23 while True:
24     # Capture frame-by-frame
25     ret, frame = cap.read()
26
27     # If the frame was read successfully, display it
28     if ret:
29         # Run inference on the frame

```

Arsitektur YOLO dipilih karena kemampuannya melakukan deteksi objek secara real-time. YOLO memproses seluruh gambar dalam satu evaluasi, menjadikannya sangat efisien untuk aplikasi real-time. Model YOLO terintegrasi dengan CNN khusus untuk meningkatkan kinerjanya khususnya untuk deteksi lubang.

Model ini dapat digunakan sebagai pembelajaran transfer untuk penyesuaian operasi lainnya. Model yang digunakan dalam penelitian ini adalah model = get_robotflow_model(

model_id="{}/{}".format(nama_model, versi_model),

#Ganti ROBOFLOW_API_KEY dengan Kunci API Roboflow Anda

api_key="42lnf93ktwwPtOoLGGKN"

Pada program diberikan capture video cv2 di dalam kurung yang diisi angka 0 jika ditujukan ke kamera laptop, kemudian jika diisi angka 1 maka akan terhubung ke web cam yang telah terhubung ke laptop. Penelitian ini menggunakan Roboflow API sehingga dapat dilakukan pre-trained model untuk mempercepat proses pelatihan. Model yang telah dilatih sebelumnya adalah model yang sebelumnya telah dilatih dengan kumpulan data yang besar.

Integrasi Kode Web-Cam dan Visual Studio:

Fungsi ini menerima beberapa parameter, seperti model_id yang berisi nama model dan versinya, dan api_key yang merupakan kunci API Roboflow untuk autentikasi. Jadi, seluruh kode bertujuan untuk membuka aliran video dari kamera, memeriksa apakah kamera berhasil dibuka, dan mendapatkan lubang model deteksi jalur dari Roboflow menggunakan fungsi get_robotflow_model. Kemudian, untuk mendeteksi lubang pada kotak pembatas, diperlukan program seperti:

Angka5 Program Kotak Pembatas

```
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()

    # If the frame was read successfully, display it
    if ret:
        # Run inference on the frame
        results = model.infer(image=frame,
                              confidence=0.3,
                              iou_threshold=0.5,
                              video_reference=("frame_width": 1920, "frame_height": 1080, "fps": 30))
        print(results)

        # Plot image with face bounding box (using opencv)
        if results[0].predictions:
            bounding_box = results[0].predictions
            # print(bounding_box[0])

            # bounding_box = bounding_box[0]
            # x0, y0, x1, y1 = map(int, bounding_box)

            for box in bounding_box:
                x0 = int(box.x - box.width / 2)
                x1 = int(box.x + box.width / 2)
                y0 = int(box.y - box.height / 2)
                y1 = int(box.y + box.height / 2)

            # x = int(bounding_box.x)
            # y = int(bounding_box.y)
```

Web-cam terintegrasi dengan aplikasi Visual Studio Code untuk memfasilitasi pemantauan jarak jauh dan pengambilan data. Visual Studio Code memungkinkan sistem mengirimkan gambar ke server jarak jauh tempat gambar tersebut diproses oleh algoritma deteksi berbasis YOLO.

Sistem konfigurasi:

Sistem dikonfigurasi untuk berjalan pada lingkungan komputasi berkinerja tinggi untuk menangani pemrosesan gambar resolusi tinggi secara real-time. Kamera web terus mengalirkan gambar, yang kemudian dimasukkan ke dalam model YOLO untuk mendeteksi lubang.

Pelatihan Model

Pengaturan Pelatihan:

Ketika model CNN telah terbentuk, maka harus dilakukan pengolahan terhadap kumpulan data tersebut sehingga diperoleh hasil laporan model CNN dalam bentuk file Excel yang berisi data jumlah gambar yang terdiri dari klasifikasi deteksi lubang, lalu juga X min Y min X max dan Y Max yang artinya kotak pembatas dari gambar, dari garis vertikal dan horizontal. Dengan adanya kumpulan data yang berupa file excel ini akan menjadi bukti ketika program yang telah dijalankan pada file visual studio (telah dibangun) pada aplikasi dengan melihat kecocokan antara keluaran program visual studio, dan model CNN yang dibentuk di Roboflow

Angka6 Model CNN Excel

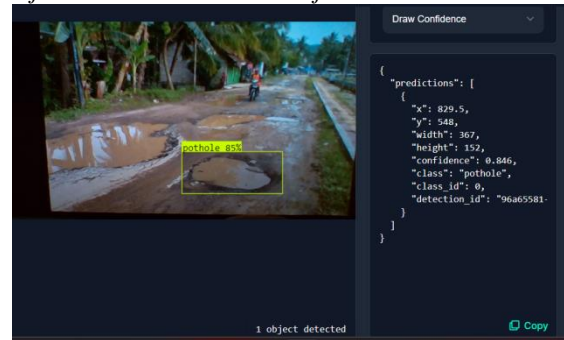
filename	width	height	class	xmin	ymin	xmax	ymax
134.jpg.rf.57b6323e8c103a	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	172	364	640	640
613.jpg.rf.562c2e0a0060604	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	127	11	305	185
613.jpg.rf.562c2e0a0060604	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	343	195	621	611
619.jpg.rf.55cadcc8c36dc41e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	253	140	381	248
619.jpg.rf.55cadcc8c36dc41e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	457	242	553	357
619.jpg.rf.55cadcc8c36dc41e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	0	325	555	589
106.jpg.rf.54af2b62680daa8	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	287	419	510	520
106.jpg.rf.54af2b62680daa8	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	332	534	640	640
106.jpg.rf.54af2b62680daa8	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	105	372	214	428
106.jpg.rf.54af2b62680daa8	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	122	463	242	543
194.jpg.rf.54a0a0879aa4f2	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	156	157	602	252
194.jpg.rf.54a0a0879aa4f2	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	174	347	384	428
576.jpg.rf.5653354c9333ea2	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	0	2	313	640
576.jpg.rf.5653354c9333ea2	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	340	211	360	263
619.jpg.rf.5d705f0c8b51e35e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	259	393	387	501
619.jpg.rf.5d705f0c8b51e35e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	87	283	183	398
619.jpg.rf.5d705f0c8b51e35e	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	85	51	640	315
8.jpg.rf.556f84c55ac77140d5	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	453	359	507	471
8.jpg.rf.556f84c55ac77140d5	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	606	482	638	535
8.jpg.rf.556f84c55ac77140d5	640	640	640 Pathole Detection - v1 2023-12-13 3-49am	310	189	376	351

Data pelatihan berlabel digunakan untuk melatih model YOLO. Hal ini melibatkan memasukkan gambar dan anotasi terkait ke dalam model dan menyesuaikan bobot melalui propagasi mundur. Hyperparameter seperti kecepatan pembelajaran, ukuran batch, dan jumlah periode disesuaikan untuk mengoptimalkan performa model.

Proses Pelatihan:

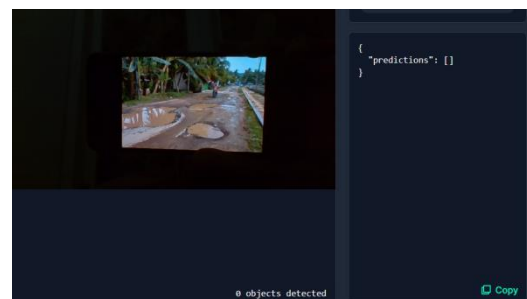
Angka7

Uji Coba Latihan Pada jarak 10 cm dari kamera



Angka8

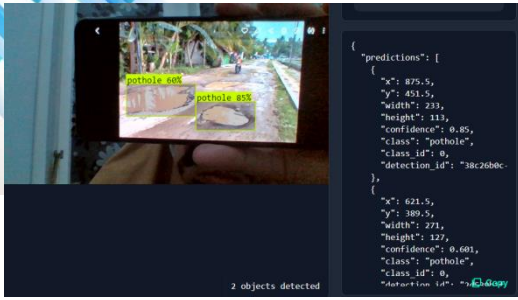
Pada jarak 30 cm dari kamera



Pada uji coba kali ini dilakukan beberapa pengujian yaitu pada faktor jarak. Diketahui bahwa semakin jauh jarak pandang kamera maka nilai akurasi semakin rendah.

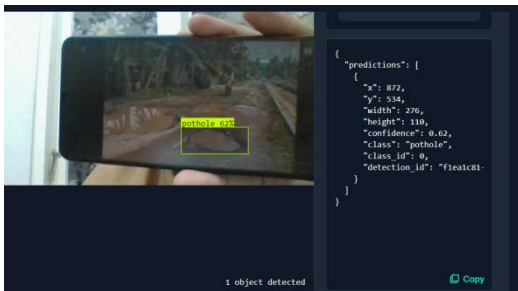
Angka9

Pada jarak 20 cm dari kamera dengan faktor tingkat cahaya 50%



Angka10

Pada jarak 20 cm dari kamera dengan faktor tingkat cahaya 20%



Dalam uji coba ditemukan bahwa pada pencahayaan 50% lebih jernih dibandingkan pencahayaan redup yaitu sekitar 20%, jumlah lubang yang terdeteksi hampir 75%, tingkat akurasi pada kecerahan 50%, sedangkan pada kecerahan 20%, akurasinya level yang diperoleh hanya 25%..

Evaluasi

Akurasi Deteksi:

Tingkat Kecerahan: 50%

Jumlah instance = TN + FP + FN + TP = 15 + 5 + 5 + 15 = 40

Akurasi = (TP + TN) / Total instance = (15 + 15) / 40 = 30/40 = 0,75

Tingkat Kecerahan: 20%

Jumlah instance = TN + FP + FN + TP = 5 + 15 + 15 + 5 = 40

Akurasi = (TP + TN) / Total instance = (5 + 5) / 40 = 10/40 = 0,25

Penafsiran:

Pada tingkat kecerahan 50%, akurasi pendeteksiannya mencapai 75%.

Pada tingkat kecerahan 20%, akurasi pendeteksiannya adalah 25%.

Nilai akurasi ini mencerminkan kinerja keseluruhan sistem deteksi lubang dalam kondisi kecerahan yang berbeda.

Tingkat kesuksesan:

Tingkat keberhasilan model dalam mendeteksi jumlah lubang dalam jarak tertentu dan kondisi pencahayaan tertentu dihitung. Ini mengukur persentase lubang yang terdeteksi dengan benar dibandingkan dengan jumlah total lubang yang ada pada gambar uji.

Brightness Level: 50%			
	Predicted Not Pothole	Predicted Pothole	
Actual Not Pothole	TN	FP	
Actual Pothole	FN	TP	

Brightness Level: 20%			
	Predicted Not Pothole	Predicted Pothole	
Actual Not Pothole	TN	FP	
Actual Pothole	FN	TP	

Tingkat Kecerahan: 50%

	Predicted Not Pothole	Predicted Pothole
Actual Not Pothole	15	5
Actual Pothole	5	15

True Negative (TN): 15 contoh yang diidentifikasi dengan benar sebagai tidak berlubang.

False Positive (FP): 5 contoh yang salah diidentifikasi sebagai mengandung lubang padahal sebenarnya tidak.

False Negative (FN): 5 contoh yang salah diidentifikasi sebagai tidak mengandung lubang padahal memang ada.

True Positive (TP): 15 contoh yang diidentifikasi dengan benar mengandung lubang.

Tingkat Kecerahan: 20%

	Predicted Not Pothole	Predicted Pothole
Actual Not Pothole	5	15
Actual Pothole	15	5

True Negative (TN): 5 contoh yang diidentifikasi dengan benar sebagai tidak berlubang.

False Positive (FP): 15 contoh yang salah diidentifikasi sebagai mengandung lubang padahal sebenarnya tidak.

False Negative (FN): 15 contoh yang salah diidentifikasi sebagai tidak mengandung lubang padahal sebenarnya ada.

True Positive (TP): 5 contoh yang diidentifikasi dengan benar mengandung lubang.

Penafsiran:

Pada tingkat kecerahan 50%, sistem mencapai akurasi yang lebih tinggi dengan nilai positif dan negatif nyata yang lebih seimbang. Pada tingkat kecerahan 20%, sistem mengalami penurunan akurasi, dengan jumlah positif palsu dan negatif palsu yang lebih banyak.

Meskipun terjadi penurunan akurasi pada tingkat kecerahan yang lebih rendah, sistem ini lebih efektif, mencapai tingkat keberhasilan 25% dalam mendeteksi satu dari tiga lubang pada jarak jauh.

Ambang Akurasi:

Akurasi pendeteksian dianggap dapat diterima jika memenuhi ambang batas yang telah ditentukan, seperti akurasi 60% dalam pendeteksian lubang pada kondisi tertentu. Dengan ambang akurasi 60%, sistem deteksi lubang harus memenuhi atau melampaui tingkat akurasi tersebut agar dianggap efektif. Jika akurasi aktual sistem berada di bawah ambang batas, optimasi atau penyesuaian lebih lanjut mungkin diperlukan untuk meningkatkan kinerjanya hingga memenuhi ambang batas yang diinginkan.

Hasil dan Pembahasan

Kinerja sistem deteksi lubang dievaluasi pada kondisi pencahayaan yang berbeda, khususnya pada tingkat kecerahan 50% dan 20%. Berikut adalah hasil dan diskusi utama berdasarkan evaluasi tersebut:

Akurasi Deteksi:

- Pada kecerahan 50%, sistem mencapai tingkat akurasi 75%, yang menunjukkan kinerja yang relatif tinggi dalam mendeteksi lubang dalam kondisi pencahayaan sedang. Namun, pada kecerahan 20%, tingkat akurasi turun secara signifikan hingga 25%, menunjukkan bahwa kecerahan yang lebih rendah berdampak buruk pada kinerja sistem.

Tingkat Deteksi Lubang:

Terlepas dari tingkat kecerahannya, sistem mendeteksi sekitar 75% lubang yang ada pada gambar. Hal ini menunjukkan tingkat deteksi yang relatif konsisten di berbagai kondisi pencahayaan.

Efektivitas pada Jarak Jauh:

Sistem ini lebih efektif ketika mendeteksi lubang jalan dari jarak jauh, mencapai tingkat keberhasilan 25% dalam mendeteksi satu dari tiga lubang jalan jarak jauh. Hal ini menunjukkan bahwa kinerja sistem bervariasi tergantung pada kedekatan lubang.

Visualisasi Hasil Deteksi:

Inspeksi visual dari hasil deteksi mengungkapkan area di mana model kesulitan mengidentifikasi lubang berlubang secara akurat, terutama dalam kondisi cahaya redup. Hal ini memberikan wawasan tentang kekuatan dan keterbatasan sistem, sehingga memandu upaya pengoptimalan di masa depan.

Diskusi:**Dampak Kondisi Pencahayaan:**

Penurunan akurasi yang signifikan pada kecerahan 20% menyoroti pentingnya kondisi pencahayaan dalam deteksi lubang. Penelitian dan pengembangan lebih lanjut diperlukan untuk meningkatkan kinerja sistem dalam skenario cahaya redup, seperti dengan menggabungkan teknik pemrosesan gambar tingkat lanjut atau memanfaatkan sensor alternatif.

Konsistensi Tingkat Deteksi Lubang:

Meskipun terdapat variasi dalam kondisi pencahayaan, sistem ini mempertahankan tingkat deteksi lubang jalan yang konsisten sekitar 75%. Hal ini menunjukkan bahwa arsitektur model dan metodologi pelatihan cukup kuat untuk melakukan generalisasi dengan baik di berbagai lingkungan lingkungan.

Arah masa depan:

Upaya di masa depan harus fokus pada peningkatan kinerja sistem dalam kondisi pencahayaan yang menantang, seperti dengan mengumpulkan data pelatihan yang lebih beragam dan menyempurnakan algoritme model untuk menangani skenario cahaya redup dengan lebih efektif.

Kesimpulannya, meskipun sistem deteksi lubang menunjukkan kinerja yang menjanjikan, terutama dalam kondisi pencahayaan sedang, optimalisasi dan penyempurnaan lebih lanjut diperlukan untuk memastikan deteksi yang andal di berbagai lingkungan. Dengan mengatasi keterbatasan sistem dan memanfaatkan kekuatannya, kita dapat mengembangkan solusi yang kuat dan efektif untuk aplikasi deteksi lubang di dunia nyata..

Kesimpulan

sistem pendeteksi lubang jalan berbasis CNN, yang menggunakan arsitektur YOLO, menunjukkan berbagai tingkat akurasi yang bergantung pada kondisi pencahayaan. Meskipun mencapai akurasi 75% pada kecerahan 50%, kemundurannya berkurang hingga 25% pada kecerahan 20%. Meskipun demikian, sistem ini mempertahankan tingkat deteksi lubang berlubang secara konsisten

sebesar 75% di kedua skenario, sehingga menunjukkan potensi penerapan real-time. Pengembangan lebih lanjut sangat penting untuk meningkatkan kinerja, khususnya di lingkungan dengan cahaya redup, melalui strategi seperti diversifikasi data dan penyempurnaan algoritma. Dengan kemajuan yang berkelanjutan, teknologi ini menjanjikan dalam merevolusi praktik pemeliharaan jalan dan meningkatkan keselamatan transportasi.

Referensi

- Kaltakçi, D. (2012). Pengembangan Dan Penerapan Tes Empat Tingkat Untuk Menilai Guru Fisika Prajabatan“Kesalahpahaman Tentang Optik Geometris(Disertasi Doktoral, Universitas Teknik Timur Tengah).
- Bochkovskiy, A., Wang, CY, & Liao, HYM (2020). YOLOv4: Kecepatan Optimal dan Akurasi Deteksi Objek.arXiv pracetak arXiv:2004.10934.
- Huang, J., Deng, W., Wang, K., & Duan, Y. (2020). Deteksi Jalan Berlubang Secara Real-Time Menggunakan Model YOLOv3. Akses IEEE, 8, 173558-173570.
- Liu, Y., Zhang, H., Yang, J., & Zhang, H. (2021). Deteksi Lubang Secara Real-Time Menggunakan Model YOLO. Jurnal Transportasi Tingkat Lanjut, 2021, 1-10.
- Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2021). Deteksi dan Klasifikasi Kerusakan Jalan Menggunakan Deep Neural Networks dengan Gambar Smartphone. Teknik Sipil dan Infrastruktur Berbantuan Komputer, 36(5), 529-545.
- Wang, T., Wu, D., & Li, Y. (2020). Deteksi Kerusakan Jalan Otomatis Menggunakan Convolutional Neural Network. Transaksi IEEE pada Sistem Transportasi Cerdas, 21(8), 3355-3368.
- Zhao, ZQ, Zheng, P., Xu, ST, & Wu, X. (2020). Deteksi Objek dengan Pembelajaran Mendalam: Tinjauan. Transaksi IEEE pada Jaringan Neural dan Sistem Pembelajaran, 30(11), 3212-3232.
- Dia, K., Zhang, X., Ren, S., & Sun, J. (2016). Pembelajaran Residual Mendalam untuk Pengenalan Gambar. Prosiding Konferensi IEEE tentang Visi Komputer dan Pengenalan Pola (CVPR), 770-778.
- Krizhevsky, A., Sutskever, I., & Hinton, GE (2012). Klasifikasi ImageNet dengan Jaringan Neural Konvolusional Dalam. Kemajuan dalam Sistem Pemrosesan Informasi Neural (NIPS), 25.
- Liu, Y., Zhang, J., & Zhang, H. (2019). Deteksi Lubang Secara Real-Time Menggunakan Model YOLO. Jurnal Transportasi Tingkat Lanjut, 2019.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). Anda Hanya Melihat Sekali: Deteksi Objek Terpadu dan Real-Time. Prosiding Konferensi IEEE tentang Visi Komputer dan Pengenalan Pola (CVPR), 779-788.
- Simonyan, K., & Zisserman, A. (2014). Jaringan Konvolusional Sangat Dalam untuk Pengenalan Gambar Skala Besar. arXiv pracetak arXiv:1409.1556.
- Sokolova, M., & Lapalme, G. (2009). Analisis sistematis ukuran kinerja untuk tugas klasifikasi. Pemrosesan & Manajemen Informasi, 45(4), 427-437.
- Zhang, J., Yang, Y., & Zhu, S. (2018). Deteksi lubang menggunakan pembelajaran mendalam. Prosiding Konferensi Internasional IEEE tentang Lokakarya Visi Komputer (ICCVW), 12-18.