IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Efficient Approach for Road Damage Detection using Deep Learning.

[1]Varun Chaudhari, [2] Prof. Vina Lomte,[3]Chirag Kumar,[4] Nidheya Labhade, [5]Parikshit Kokare.

[1]B.E. Computer Engineer, [2]Project Guide and Head of Computer Engineering Department, [3]B.E. Computer Engineer, [4]B.E. Computer Engineer, [5]B.E. Computer Engineer

[1]Department of Computer Engineering,

[1]RMD Sinhgad School of Engineering, SPPU, Pune, India.

Abstract: Most of the rural or the urban municipalities and road authorities have a hard times mapping the surface damages caused due to various reasons such as heavy rains, natural calamities, or other factors that lead to cracks and holes appearing on the surface of the roads. These organizations or private entities look out for solutions to implement automated methods of reporting damages on the surface of the road. In most cases, they lack the technology required for the purpose of mapping the damages on the roads. One of the biggest problems for commuters is that, they have to face a lot of damaged sections of road which leads to frequent reduction of the speed at which they travel, wasting a lot of time and effort from the rider's perspective, which thereby increases the travel time to their destinations. Damage to the road can be fatal many times when traveling at higher speed and all of a sudden meeting a damaged part of the road. Moreover, it has the potential to detect frequent congestion and the reason behind it and also propose a solution for the same. Most of the time, these congestion are caused by damage to the road making the commuters travel at a much lesser speed than what is recommended.

Keywords: **Convolutional Neural Network(CNN), Smart Road Damage Detection, Classification, Machine Learning, Image Segmentation, Pre-processing, Segmentation, Feature Extraction.**

I.INTRODUCTION

Road surface damage detection and warning systems are essential for maintaining road safety, cutting down on maintenance expenses, and enhancing the effectiveness of transportation as a whole. These systems are made to recognize different types of surface damage to the road, like potholes, cracks, and deteriorations, and to alert drivers, road maintenance agencies, or self-driving cars on time. Systems for detecting and alerting users to damage to the road surface are crucial for keeping road networkseffective and safe. To identify damage and notify drivers and authorities on time, they use a combination of sensor technologies, machine learning, and communication systems. These systems are anticipated to become more crucial to the management of transportation infrastructure as technology advances.

To do this, a system that can identify road flaws should be created. The most common causes of traffic accidents are potholes and speed bumps. It should be recognized and relayed to the other vehicle.

II. LITERATURE SURVEY

Methods and techniques used by Referenced Papers.

The below table gives the detail survey on the techniques and methods used in different phases of Road Damage Detection. It also specifies the accuracy and outcome of the proposed system of the referenced paper.

Sr. No.	Publication Detail	Methods Used	Dataset Used	Accuracy	Research Gaps Identified
1	Optimal Fuzzy Wavelet Neural Network Based Road Damage Detection MOHAMMAD ALAMGEER, HEND KHALID ALKAHTANI. (IEEE 2023)	CNN	Kaggle	98.56%	The simulation values demonstrate the supremacy of the OFWNN-RDD algorithm over other models. The performance of the OFWNN-RDD method will be enhanced by the ensemble learning process.

2	Automated Road Damage Detection Using UAV Images and Deep Learning Techniques. Luís Augusto Silva, Valderi Reis Quietinho Leithardt (IEEE2023).	YOLO	Kaggle	87%	An architecture capable of detecting road damage and demonstrated that new architecture versions, such as YOLOv5 and YOLOv7, can improve upon previous work. The different types of images, such as multispectral images and LIDAR sensors, to further enhance the performance.
3	A Deep Learning Model for Road Damage Detection After an Earthquake Based on Synthetic Aperture Radar (SAR) and Field Datasets Sadra Karimzadeh , (IEEE 2022).	MLP	Kaggle	89%	Collecting IRI data with traditional methods is a costly and time-consuming process; therefore, it is difficult to use this technique in practical crisis management and road monitoring.
4	Road Damage Detection From Post-Disaster High-Resolution Remote Sensing Images Based on TLD Framework KANG ZHAO 1,4, JINGJING LIU 2,5, (IEEE 2022).	CNN	Kaggle	90%	The proposed damaged road region detection method based on road tracking detection has great potential in road damage detection. In the future, the proposed method will be adapted to assess the road damage level assessment.
5	Pothole Detection Based on Disparity Transformation and Road Surface Modeling Rui Fan, Member, IEEE, Umar Ozgunalp (IEEE 2021).	CNN	Kaggle	87%	The road surface without the obvious defects improve the speed of the vehicle as much as possible to ensure the efficiency of life and production.
6	Detection of Road Condition Defects Using Multiple Sensors and IoT Technology: A Review A. ALRAJHI1, K. ROY 2, L. QINGGE 2 B. (ITC 2022)	CNN	Kaggle	89%	On the other hand, mobile sensing has led to AVs, which according to reviewed literature, will transform the transportation sector. Nevertheless, the extent of their effect on transportation infrastructure is not yet fully understood.
7	Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning SEUNGBO SHIM 1,2 AND GYE-CHUN CHO3 [IEEE-2020]	RF	Kaggle	90%	The strategy was evaluated with respect to four performance indexes to determine its accuracy compared to the existing method. This is the result of the experiments on images that show general road-surface damage.
8	Evolving Pre-TrainedCNN Using Two-Layer Optimizer for Road Damage Detection From Drone Images. HUSSEIN SAMMA, SHAHREL AZMIN SUANDI 3 (IEEE 2021)	CNN	Kaggle	87%	In future work, it is suggested that the proposed model could be evaluated using a large dataset with heterogeneous road damage types such as crack, falling block, and leakage.

9	Yet Deep Learning Approach for Road Damage Detection using Ensemble Learning Vinuta Hegde, Dweep Trivedi, Abdullah Alfarrarje, Aditi Deepak, Seon Ho Kim, Cyrus Shahabi.	CNN	Kaggle	88%	An automated solution for road damage detection and classification using image analysis is nowadays needed for smart city applications. Nevertheless, the extent of their effect on transportation infrastructure is not yet fully understood.
10	Study on the Effect of Electrical Conductivity on Road Surface Wetness Classification Using Capacitive Measurement Data Jakob Döring , Lukas Lütkehaus (IEEE -2022)	CNN	Kaggle	90%	To this end, we generated reproducible spray water scenarios representing a specific wheel speed and a road surface wetness class.
11	Attention Based Coupled Framework for Road and Pothole Segmentation Shaik Masihullah, Ritu Garg, Prerana Mukherjee (IEEE 2020)	CNN	Kaggle	87%	It also provides audio and tactile feedback to the user for secure movement on the road.
12	ROAD DEFORMATION DETECTION Kundana Angelina Govada , Dr. K Suvarna Vani [IEEE-2020]	YOLO	Kaggle	89.30%	High-quality and large-scale datasets are essential for training and evaluating road damage detection models.
13	A Comparative Evaluation of the Deep Learning Algorithms for Pothole Detection Roopak Rastogi, Uttam Kumar[IEEE-2020]	CNN	Kaggle	94.20%	Comparative evaluations should consider a wide range of evaluation metrics, not just accuracy.
14	Real-time road pothole mapping based onvibration analysis in smart city Dong Chen , Member, IEEE, Nengcheng Chen , Xiang Zhang , and YuhangGuan.	CNN	Kaggle	90.04%	It conveniently monitor urban roads using various cameras, such as surveillance cameras, in-vehicle cameras, or smartphones, and recognize their conditions by detecting specific types of road damages in order to plan maintenance resources efficiently based on the identified spots .
15	Global Road Damage Detection: State-of-the-art Solutions Deeksha Arya*,1,2 , Hiroya Maeda2 [IEEE-2020]	CNN	Kaggle	84.60%	High-quality and large-scale datasets are essential for training and evaluating road damage detection models.
16	Generative adversarial network for road damage detection Hiroya Maeda Takehiro Kashiyama [Computer-Aided Civil-2020]	CNN	Kaggle	86.96%	Road damage data is often highly imbalanced, with very few instances of damage compared to undamaged road sections.

17	Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN Pascal Fassmeyer, Felix Kortmann [IEEE-2021]	YOLO	Kaggle	80%	Autonomous vehicles operate in dynamic and complex real-world environments.
18	Deep Learning-based Road Damage Detection and Classification for Multiple Countries Deeksha Arya*1,2, Hiroya Maeda [published by Elsevier-2021]	YOLO	Kaggle	89.53%	Manually annotating road damage data for multiple countries is time-consuming and expensive.
19	Transfer Learning-based Road Damage Detection for Multiple Countries Deeksha Arya∗1,2, Hiroya Maeda [IEEE-2022]	YOLO-CNN	Kaggle	85%	Road damage types and patterns can vary due to cultural, environmental, and geographical factors.
20	Road Damage Detection and Classification with Detectron2 and Faster R-CNN Vung Pham , Chau Pham [IEEE-2020]	CNN	Kaggle	95%	Road damage is not limited to a few common categories; it can include various types of cracks, potholes, and other defects.

III. ALGORITHMIC SURVEY

1) CNN: Artificial Intelligence has been rapidly expanding to close the gap between human and machine capabilities. To achieve incredible things, researchers and enthusiasts alike work on many facets of the field. Computer vision is one of these many domains. The goal of this field is to make it possible for machines to see and understand the world similarly to humans, and to even use that ability for a wide range of applications, including Natural Language Processing, Image Video Recognition, Media Recreation, Recommendation Systems, and Image Analysis Classification. Convolutional Neural Networks are the main algorithm that has been used to build and refine the advances in computer vision with deep learning over time.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can recognize different objects and aspects in an input image and distinguish between them by assigning weights and biases that can be learned. A ConvNet requires a lot less pre-processing than other classification algorithms. While filters are manually designed in more archaic techniques, ConvNets can learn these filters and characteristics given sufficient training.

ConvNet architecture was inspired by the structure of the visual cortex and is comparable to the connectivity pattern of neurons in the human brain. Individual neurons respond to stimuli only in the Receptive Field, a tiny portion of the visual field. A collection of such fields overlap to cover the entire visual area.

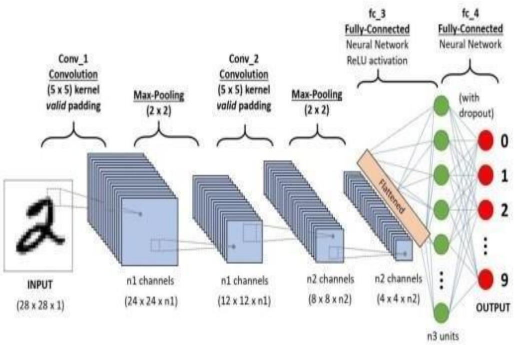

Figure 5.1: CNN Sequence

2) Hyperparameter Optimization: The problem of selecting a set of ideal hyperparameters for a learning algorithm is known as hyperparameter tuning in machine learning. A hyperparameter is a parameter that governs the learning process through its value. On the other hand, other parameters, usually node weights, have learned values.

To generalize various data patterns, the same type of machine learning model may need to use different weights, constraints, or learning rates. These measurements, referred to as hyperparameters, need to be adjusted in order for the model to solve the machine-learning problem as best it can. Using provided independent data, hyperparameter optimization determines a pair of hyperparameters that produce an ideal model that minimizes a predetermined loss function. The associated loss is returned by the objective function after receiving a tuple of hyperparameters. To estimate this generalization performance and determine the set of hyperparameter values that maximize it, cross-validation is frequently employed.

3) Grid search: Grid search, also known as a parameter sweep, is the conventional method of executing hyperparameter optimization. It consists of a thorough search across a manually specified subset of a learning algorithm's hyperparameter space. Some performance metrics, usually assessed by evaluation on a hold-out validation set or cross-validation on the training set, must drive a grid search algorithm.

Before using grid search, manually set bounds and discretization may be required because the parameter space of a machine learner may contain real-valued or unbounded value spaces for some parameters.

For example, a typical soft-margin SVM classifier equipped with an RBF kernel has at least two hyperparameters that need to be tuned for good performance on unseen data: a regularization constant C and a kernel hyperparameter γ. Both parameters are continuous, so to perform a grid search, one selects a finite set of "reasonable" values for each, say

�∈{10,100,1000}

�∈{0.1,0.2,0.5,1.0}

Grid search then trains an SVM with each pair (C, γ) in the Cartesian product of these two sets and evaluates their performance on a held-out validation set (or by internal cross-validation on the training set, in which case multiple SVMs are trained per pair). Finally, the grid search algorithm outputs the settings that achieved the highest score in the validation procedure.

Grid search suffers from the curse of dimensionality, but is often embarrassingly parallel because the hyperparameter settings it evaluates are typically independent of each other.

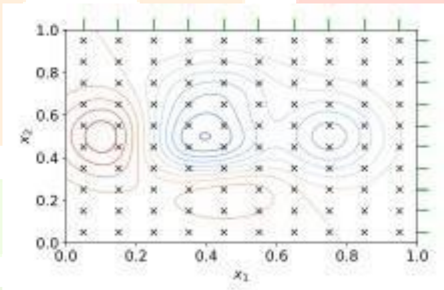

Figure 5.2: Grid Search

Grid search across different values of two hyperparameters. For each hyperparameter, 10 different values are considered, so a total of 100 different combinations are evaluated and compared. Blue contours indicate regions with strong results, whereas red ones show regions with poor results.

Due to the possibility that the parameter space of a machine learner contains real-valued or unbounded value spaces for some parameters, manually setting bounds and discretization may be necessary prior to using grid search.

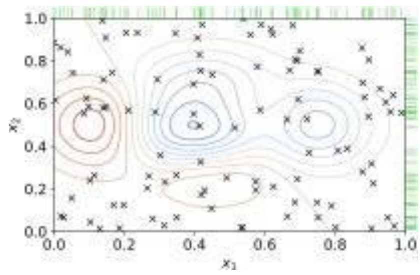

Figure 5.3: Grid Search Traditional Sequence

Random search across different combinations of values for two hyperparameters. In this example, 100 different random choices are evaluated. The green bars show that more individual values for each hyperparameter are considered compared to a grid search.

4) Bayesian optimization: A global optimization technique for noisy black-box functions is called Bayesian optimization. When Bayesian optimization is used for hyperparameter optimization, a probabilistic model of the function mapping hyperparameter values to the objective assessed on a validation set is constructed. Bayesian optimization seeks to collect data revealing as much as possible about this function and, specifically, the location of the optimum by iteratively evaluating a promising hyperparameter

configuration based on the current model and then updating it. It attempts to strike a balance between exploitation (hyperparameters expected close to the optimum) and exploration (hyperparameters for which the outcome is most uncertain). In practice, Bayesian optimization has been shown to obtain better results in fewer evaluations compared to grid search and random search, due to the ability to reason about the quality of experiments before they are run.

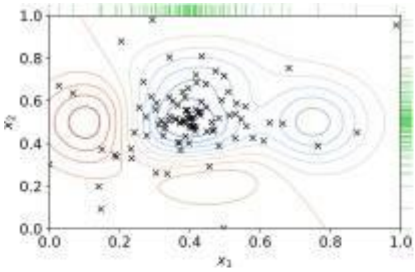

Figure 5.4: Bayesian optimization

Methods such as Bayesian optimization smartly explore the space of potential choices of hyperparameters by deciding which combination to explore next based on previous observations.

5) Gradient-based optimization: For certain learning algorithms, the gradient with respect to hyperparameters can be computed, and the hyperparameters can then be optimized via gradient descent. Neural networks were the primary focus of these techniques' initial application. Subsequently, these techniques have been expanded to encompass additional models like logistic regression and support vector machines.
An alternative method for obtaining a gradient with respect to hyperparameters is to use automatic differentiation to differentiate the steps of an iterative optimization algorithm. A more recent study in this area estimates a stable approximation for the inverse Hessian and computes hyper gradients using the implicit function theorem. The technique needs continuous memory and scales to millions of hyperparameters.
An alternative method involves training a hypernetwork to approximate the optimal response function. This method's ability to handle discrete hyperparameters is one of its benefits. A memory-efficient variant of this method is provided by self-tuning networks, which select a compact representation for the hypernetwork. Lately, Δ-STN has enhanced this technique even more by slightly altering the hypernetwork's parameters, which expedites the training process. Additionally, by linearizing the network in the weights and eliminating needless nonlinear effects from large changes in the weights, Δ-STN produces a better approximation of the best-response Jacobian.
In addition to using hypernetwork techniques, gradient-based methods can also be applied to continuous parameter relaxation in order to optimize discrete hyperparameters. Such methods have been extensively used for the optimization of architecture hyperparameters in neural architecture search.

IV. METHODOLOGY

1. Introduction:

The system is designed using the concepts of machine learning and artificial intelligence. The system GUI was designed using the Tkinter library to make the login page. The login page is connected to the database using the MySQL connector. The database is basically used for storing user details like Username. The tool used for database functionalities was MySQL GUI Browser.

1.1 Database Requirements: SQLITE

1.2 Software Requirements (Platform Choice):

Operating system: Windows 7 or more.
Coding Language: Python
IDE: Spyder

1.3 Hardware Requirements:

System: Intel I3 Processor and above.
Hard Disk: 20 GB
Ram: 8GB

2. Software Information:

Python: Python is a general-purpose, interpreted, high-level programming language. Python was developed by Guido van Rossum and was first made available in 1991. Its design philosophy places a strong emphasis on code readability and makes extensive use of whitespace. Its object-oriented methodology and language constructs are designed to assist programmers in writing logical, understandable code for both small and large-scale projects.
Python uses garbage collection and dynamic typing. It is compatible with various programming paradigms, such as object-oriented, functional, and structured (especially procedural). Because of its extensive standard library, Python is frequently referred to as a "batteries included" language.

The ABC language was replaced with Python in the late 1980s. With the release of Python 2.0 in 2000, features like reference counting in the garbage collection system and list comprehensions were added.
Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.
The Python 2 language was officially discontinued in 2020 (first planned for 2015), and ”Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release.” No more security patches or other improvements will be released for it. With Python 2’s end-of-life, only Python 3.6.x and later are supported.

Anaconda: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.
Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small,bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.
Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).
The big difference between conda and the pip package manager is in how package dependencies are managed, which is asignificant challenge for Python data science and the reason conda exists.
When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpylibrary than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.
In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.
Open-source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or theuser’s own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anythingavailable on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself andwhat pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploadingthem to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

Spyder: Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.
Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your own software.

2.1 Features:

-Editor:

Work efficiently in a multi-language editor with a function/class browser, real-time code analysis tools (pyflakes, pylint, and pycodestyle), automatic code completion (jedi and rope), horizontal/vertical splitting, and go-to-definition.

-Interactive console:

Harness the power of as many IPython consoles as you like with full workspace and debugging support, all within the flexibility of a full GUI interface. Instantly run your code by line, cell, or file, and render plots right in line with the output or in interactive windows.

-Documentation viewer:

Render documentation in real-time with Sphinx for any class or function, whether external or user-created, from either the Editor or a Console.

-Variable explorer:

Inspect any variables, functions or objects created during your session. Editing and interaction is supported with many common types, including numeric/strings/bools, Python lists/tuples/dictionaries, dates/timedeltas, Numpy arrays, Pandas index/series/dataframes, PIL/Pillow images, and more.

-Development tools:

Examine your code with the static analyzer, trace its execution with the interactive debugger, and unleash its performance with the profiler. Keep things organized with project support and a built-in file explorer, and use find in files to search across entire projects with full regex support.

V. CONCLUSION AND FUTURE SCOPE

Road surface damage detection is an essential component of managing transportation infrastructure, with broad consequences for sustainability, efficiency, and safety. To ensure the safety of all road users, prolong the life of road infrastructure, and maximize resource allocation, it is imperative to recognize and address various forms of damage on road surfaces, such as potholes, cracks, and rutting. It is essential to the general health of the economy, the environment, and communities because it makes sure that road networks are long-lasting, safe, and effective. Future road surface damage detection techniques should see more advancements and improvements as technology develops, making transportation systems even safer and more dependable.

VI. REFERENCES

[1] Optimal Fuzzy Wavelet Neural Network Based Road Damage Detection Mohammad Alamgeer1, Hend Khalid Alkahtani 2, (Ieee 2023) Doi: 10.1109/Access.2023.3283299.

[2] Automated Road Damage Detection Using Uav Images And Deep Learning Techniques Luís Augusto Silva 1, Valderi Reis Quietinho Leithardt (Ieee 2023). Digital Object Identifier:10.1109/Access.2023.3287770.

[3] A Deep Learning Model For Road Damage Detection After An Earthquake Based On Synthetic Aperture Radar (Sar) And Field Datasets Sadra Karimzadeh , (Ieee 2022) Doi: 10.1109/Jstars.2022.3189875.

[4] Road Damage Detection From Post-Disaster High-Resolution Remote Sensing Images Based On Tld Framework Kang Zhao 1,4, Jingjing Liu 2,5, (Ieee 2022) Digital Object Identifier: 10.1109/Access.2022.3169031.

[5] Pothole Detection Based On Disparity Transformation And Road Surface Modeling Rui Fan, Member, Ieee, Umar Ozgunalp (Ieee 2021) Doi 10.1109/Tip.2019.2933750 .

[6] Detection Of Road Condition Defects Using Multiple Sensors And Iot Technology: A Review Alrajhi1, K. Roy 2, L. Qingge 2 Digital Object Identifier 10.1109/Ojits.2023.3237480.

[7] Lightweight Semantic Segmentation For Road-Surface Damage Recognition Based On Multiscale Learning Seungbo Shim 1,2 And Gye-Chun Cho3 [Ieee-2020] Doi: 10.1109/Access.2020.299842.

[8] Evolving Pre-Trained Cnn Using Two-Layers Optimizer For Road Damage Detection From Drone Images Hussein Samma, Shahrel Azmin Suandi 3 (Ieee 2021) Doi:10.1109/Access.2021.3131231.

[9] Yet Another Deep Learning Approach For Road Damage Detection Using Ensemble Learning (Ieee 2020)Doi:10.1109/Bigdata50022.2020.9377833

[10] Study On The Effect Of Electrical Conductivity On Road Surface Wetness Classification Using Capacitive Measurement Data Jakob Döring , Lukas Lütkehaus (Ieee -2022).

[11] Attention Based Coupled Framework For Road And Pothole Segmentation Shaik Masihullah∗ , Ritu Garg† , Prerana Mukherjee(Ieee 2020) Doi: 10.1109/Icpr48806.2021.9412368 .

[12] Road Deformation Detection Kundana Angelina Govada , Dr. K Suvarna Vani [Ieee-2020] 978-1-7281-7223-1/20/$31.00 .

[13] A Comparative Evaluation Of The Deep Learning Algorithms For Pothole Detection Roopak Rastogi, Uttam Kumar[Ieee-2020] Doi: 10.1109/Indicon49873.2020.9342558 .

[14] Real-Time Road Pothole Mapping Based On Vibration Analysis In Smart City Dong Chen , Member, Ieee(2022).

[15]Global Road Damage Detection: State-Of-The-Art Solutions Deeksha Arya*,1,2 , Hiroya Maeda2 [Ieee-2020]

[16] Generative Adversarial Network For Road Damage Detection Hiroya Maeda Takehiro Kashiyama [Computer-Aided Civil-2020] Doi: 10.1111/Mice.12561

[17]Towards A Camera-Based Road Damage Assessment And Detection For Autonomous Vehicles: Applying Scaled-Yolo And Cvae-Wgan Pascal Fassmeyer, Felix Kortmann [Ieee-2021]

[18] Deep Learning-Based Road Damage Detection And Classification For Multiple Countries Deeksha Arya*1,2, Hiroya Maeda [Published By Elsevier-2021]

[19] Transfer Learning-Based Road Damage Detection For Multiple Countries Deeksha Arya∗1,2, Hiroya Maeda [Ieee-2022]

[20] Road Damage Detection And Classification With Detectron2 And Faster R-Cnn Vung Pham , Chau Pham [Ieee-2020] Doi: 10.1109/Bigdata50022.2020.9378027