CAR DAMAGE DETECTION USING MACHINE LEARNING

Vijay M, Department of Information Technology
Vijay V, Department of Information Technology
Jegan L, Department of Information Technology
B.RAMA.M.E., Associate professor, Department of Information Technology
M.A.M College of Engineering and Technology
Siruganur ,Trichy

Abstract – Modern car insurance industries waste a lot of resources due to claim leakages, which determines the amount they pay. Currently,visual Inspections and Validations are done manually,which can delay the claim processes.Previous study have shown that classifying images is possible with a small data set,by transferring and re purposing knowledge from models trained for a different task. Our goal is to build a Car Damage classifier using a deep learning model that is able to detect the different damage types and give an accurate depiction given a car image. However, due to the limiting set of data, it can be result in being a determining factor.Training a Convolutional Network from scratch (with random initialization) is difficult because it is relatively rare to have a large enough dataset.In this project we explore the problem of classifying images containing damaged cars to try and assess the monetary value of the damage. Because of the nature of this problem,classifying this data may prove to be a difficult task since no standardized dataset exists and some of the clases utilized might not be discriminative enough. Utilizing a pretrained YOLOv8 model,we trained a classifier in order to categorize the dataset,testing 3 different cases: damaged or not (damage vs whole),damage location (front vs rear vs side),damage level (minor vs moderate vs severe).

Index Terms - YOLO model,CNN

I.INTRODUCTION

In the automotive industry,vehicle maintenance is crucial to ensure optimal and safe operation.Car body damage is a significant concern as it can diminish the vehicle's resale value and compromise driver safety. Traditionally, damage detection has relied on visual inspections or specialized measuring devices, but these methods are time-consuming and require expertise for interpretation. Recently, deep learning algorithms in computer vision have shown remarkable proficiency in object detection and classification. Hence, this study aims to utilize a deep learning algorithm for car body damage detection through image analysis. Deep learning models like YOLO and Faster R-CNN exhibit impressive performance in identifying different forms of car body damage. Car body damage is a prevalent issue that can impact vehicle functionality. Swift and precise identification of such damage can expedite repair processes and mitigate associated costs. Extensive research has been conducted to develop algorithms leveraging deep learning for car body damage detection. Among these algorithms,You Only Look Once (YOLO) stands out.YOLO is a real-time object detection algorithm capable of recognizing various objects in images and videos.It employs convolutional neural networks (CNNs) for object detection.In the context of car body damage detection,the YOLO algorithm can discern different types of damage such as scratches,dents and fractures on various car body components.This outlines a technique for identifying scratches, fractures, and other harm to car exteriors through machine learning methods.The proposed process includes gathering and preparing a dataset of car images,then processing and segmenting them to isolate the affected areas. Following this,sophisticated learning algorithms are employed to construct detection models, which are then trained and assessed using a test dataset.This suggested approach is expected to provide a more efficient and economical solution for inspecting automobile bodies,particularly relevant for sectors like taxi services that require regular vehicle checks.It has the potential to decrease time and expenses while enhancing the accuracy and reliability of inspections by automating the process.Furthermore,the findings of this study could be utilized to improve the safety and maintainability of various types of vehicles,including buses,trucks and personal cars.

DETECTION ALGORITHM FRAMEWORK

1. Mask RCNN Algorithm

The vehicle-damage-detection and segmentation system based on the Mask RCNN model.

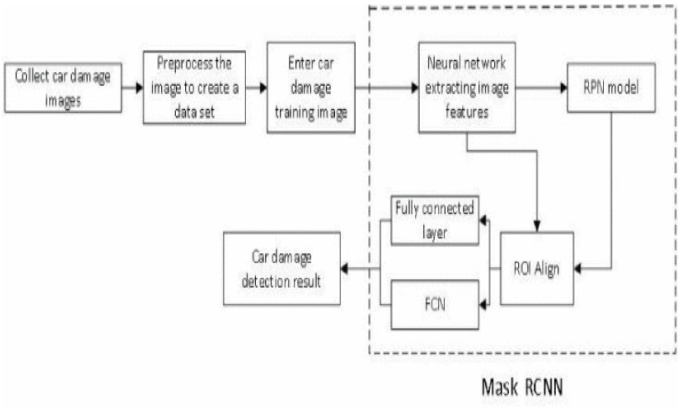

Fig 1-Mask RCNN

Mask RCNN,an expansion of Faster RCNN,is a framework designed for segmenting instances.It operates in two distinct stages: initially, it examines the image and produces the proposal, and subsequently, it classifies the proposal while generating both the bounding box and the mask.

I. Feed the image into a pre-trained ResNet 50+ FPN network model to extract features and acquire corresponding feature maps.
II. These feature maps generate numerous potential regions of interest (ROI) via RPN. Then, a softmax classifier discerns foreground and background through binary classification. Frame regression refines the positional data of these potential regions, and non-maximum suppression eliminates redundant ROIs.
III. The feature map and the remaining ROIs proceed to the RoIAlign layer, where each ROI generates a standardized feature map.
IV. Finally, the process diverges into two paths: one path enters a fully connected layer for object classification and frame regression, while the other enters a fully convolutional network (FCN) for pixel segmentation.

METHODOLOGY USED

1) YOLO-V8

The You Only Look Once (YOLO-VERSION 8) algorithm is a powerful tool in deep learning, designed for accurately detecting objects in both images and videos. It achieves this by combining deep convolutional architecture with real-time spatial analysis techniques, allowing for swift object detection. YOLO operates by utilizing a Convolutional Neural Network (CNN) to identify objects. It accomplishes this by breaking down an image into a grid of squares and then predicting the presence of objects within each square, a process known as grid cell classification. Additionally, YOLO employs bounding box regression to precisely determine the location of objects within these squares. Essentially, YOLO calculates the likelihood of an object belonging to a particular class and its precise spatial coordinates within each grid, utilizing mathematical calculations.

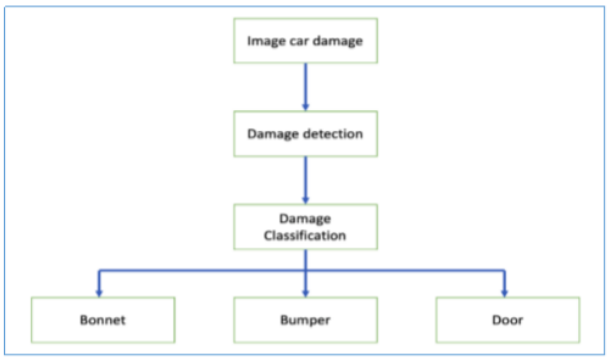

Fig 2-YOLO

2) Feature Extraction:

i. YOLOv8 uses a deep convolutional neural network (CNN) as its backbone, typically based on Darknet or ResNet architecture.
ii. The CNN extracts features from the input image at multiple scales using convolutional layers and pooling operations.
iii. Feature maps from different layers of the network are combined using feature pyramid networks (FPN) and skip connections to capture both high-level and low-level features.

3) Detection Head:

i. The feature maps are processed by the detection head, which predicts bounding boxes, confidence scores, and class probabilities for objects within each grid cell.
ii. For each grid cell, the detection head predicts multiple bounding boxes (anchors) with associated confidence scores, indicating the presence of an object and the accuracy of the bounding box.
iii. Each bounding box is associated with a class probability distribution, representing the likelihood of different object classes within the box.

4) Non-Maximum Suppression (NMS):

i. After prediction, a non-maximum suppression algorithm is applied to remove redundant bounding boxes and retain only the most confident ones for each object.
ii. Bounding boxes with confidence scores below a certain threshold are discarded, while highly overlapping boxes are merged into a single detection.

5) Output Generation:

i. The final output of the YOLOv8 algorithm consists of a list of bounding boxes, each associated with a class label and a confidence score.
ii. These bounding boxes represent the detected objects in the input image, including their location, size and class.

6) Training

i. YOLOv8 is trained using labeled training data, where each object in the image is annotated with a bounding box and a class label.
ii. The model is trained using backpropagation and optimization algorithms to minimize the difference between predicted and ground truth bounding boxes and class probabilities.

Fine-tuning and Optimization

i. YOLOv8 can be fine-tuned using transfer learning techniques on specific tasks, such as car damage detection.
ii. Hyperparameters such as learning rate, batch size and network architecture can be adjusted to optimize performance for the target task.

BINARY MASKS PROCESS

Binary masks play a crucial role in object detection and segmentation tasks, including our car damage detection system. These masks provide a pixel-level representation of the detected objects, allowing for precise localization and analysis. In our context, a binary mask is a two-dimensional array where each pixel is assigned a value of 0 or 1, indicating whether it belongs to the detected object or not. In our system, binary masks are generated to highlight the exact location of car damages within test images. Each damage type, such as dents, scratches, or cracks, is represented by a separate binary mask, allowing us to isolate and analyze individual damage instances. These masks serve as visual aids, providing a clear and intuitive representation of the detected damages. By extracting the damage locations using binary masks, we can precisely quantify the extent and severity of damages, facilitating further analysis and decision-making processes. Additionally, binary masks enable us to overlay the detected damages onto the original images, providing valuable insights for insurance claim processing, vehicle maintenance, and safety assessment. Overall, binary masks are essential components of our car damage detection system, offering a detailed and accurate representation of detected damages and enhancing the effectiveness of our automated detection process.

1. **Damage Detection:** After processing the test input images using the trained YOLOv8 algorithm,we obtain bounding box coordinates and class probabilities for each detected damage instance.These detections include various types of damages such as dents, scratches, cracks and others.
2. **Binary Mask Generation:** For each detected damage instance, we create a binary mask that corresponds to the exact location of the damage within the image.The binary mask is a two-dimensional array where each pixel is assigned a value of 1 if it belongs to the damage area and 0 otherwise.
3. **Pixel-level Classification:** To generate the binary mask, we classify each pixel within the bounding box as either part of the damage or background.This classification is based on the confidence scores and class probabilities obtained from the YOLOv8 detection.

FLASK FRAMWORK

Flask is a lightweight and flexible web framework for Python, designed to make web development simple and scalable.It provides developers with the tools they need to build web applications quickly and efficiently, without imposing too many constraints or dependencies. At its core, Flask is based on the WSGI (Web Server Gateway Interface) standard, making it compatible with a wide range of web servers and deployment options. One of Flask's key features is its simplicity. It follows the "microframework" philosophy, which means it focuses on doing one thing well – handling HTTP requests and responses – while leaving other tasks like database management and authentication to extensions or third-party libraries.This minimalistic approach gives developers the freedom to choose the components they need for their projects, resulting in lean and efficient web applications. Additionally, Flask offers a rich ecosystem of extensions and plugins that extend its functionality and provide additional features such as form validation, user authentication, and database integration. These extensions, combined with Flask's intuitive API and extensive documentation,make it easy for developers to customize and scale their applications according to their specific requirements.Whether you're building a simple RESTful API or a complex web application,Flask provides the flexibility and versatility to meet your needs.Its lightweight nature and minimalistic design make it an ideal choice for prototyping, rapid development and building scalable web solutions.

ACCURACY

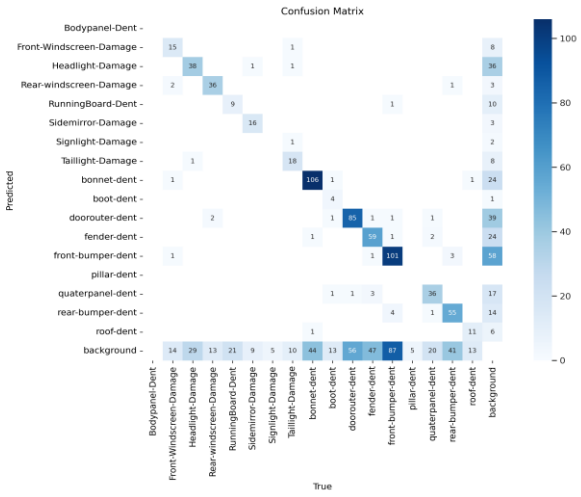

Fig – 3 MATRIX FOR DEFECTS

MODULES DESCRIPTION

1. Image dataset collection

This module involves gathering a diverse dataset of car images containing various types of damages, including dents, scratches, cracks, glass shatters, flat tires and broken lamps. The dataset encompasses different scenarios and severity levels to ensure the model's robustness and generalization capability.

2. Dataset Split

We split images in CarDD into the training set (2816 images, 70.4%), validation set (810 images, 20.25%), and test set (374 images, 9.35%) to make sure that instances of each category kept a ratio of 7:2:1 in the training, validation and test sets.To avoid data leakage,we take care to minimize the chance of near-duplicate images existing across splits by explicitly removing near-duplicates.

3. Image pre-processing

In this module, preprocess the collected images to ensure uniformity and quality before training the model. Preprocessing techniques include resizing, normalization, and augmentation to standardize the images and increase the diversity of the dataset, improving the model's performance and generalization ability.

4. YOLOv8 Algorithm training

Using the preprocessed dataset, we train the YOLOv8 algorithm, a state-of-the-art object detection model, to detect car damages from images. The algorithm learns to recognize specific damage patterns and localize damages accurately within the images.

5. Data classification

During training, we classify the data into two classes: damaged and undamaged. This classification allows the model to learn the distinguishing features of each class and accurately differentiate between damaged and undamaged car images.

6. Model build

After training, the model is built and stored in the database for future use. The database contains the trained parameters and configurations of the YOLOv8 model, enabling easy retrieval and deployment.

7. Test input image

In this module, user to upload test input images to the system for evaluation.These images may include synthetic and real-world data to assess the model's performance in different scenarios.

8. Testing and analysis

The uploaded test images are processed by the trained model, and the system evaluates its performance using metrics such as precision, recall, and F1-score.This analysis provides insights into the model's accuracy and effectiveness in detecting car damages.

9. Damage detection

Using the trained model, the system detects various types of car damages, including dents, scratches, cracks, glass shatters, flat tyres and broken lamps from the test images.

10. Mask exact damage location

The system generates binary maps that precisely highlight the location of damages within the test images.These masks enable us to extract and visualize the exact damage locations, aiding in further analysis and decision-making processes.

FLOW DIAGRAM

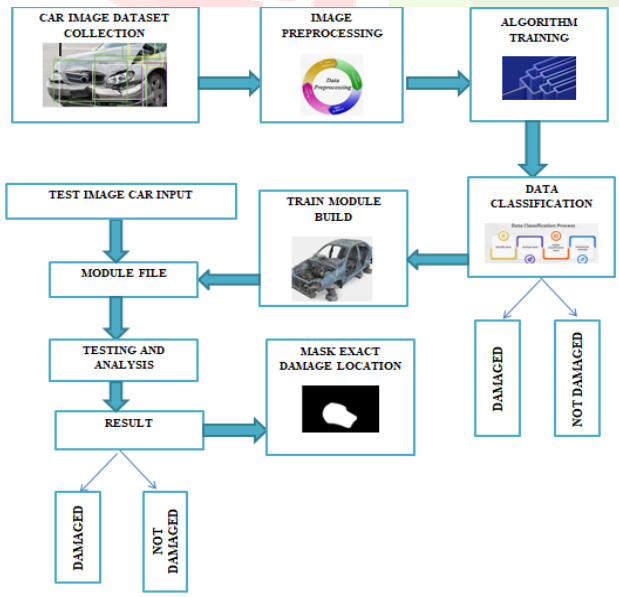

Fig – 4 BLOCK DIAGRAM

RESULT

Our experimental results demonstrate the effectiveness of the YOLOv8 algorithm with binary masking for car damage detection. The model achieves high average precision and IoU scores, indicating its ability to accurately detect and localize car damage instances across different classes. The qualitative examples further illustrate the robustness of the model in identifying various types of damage, highlighting its potential for real-world applications in insurance claims processing, accident analysis, and vehicle safety inspection.

SAMPLE OUTPUT:

Fig – 5 INPUT RAW IMAGE

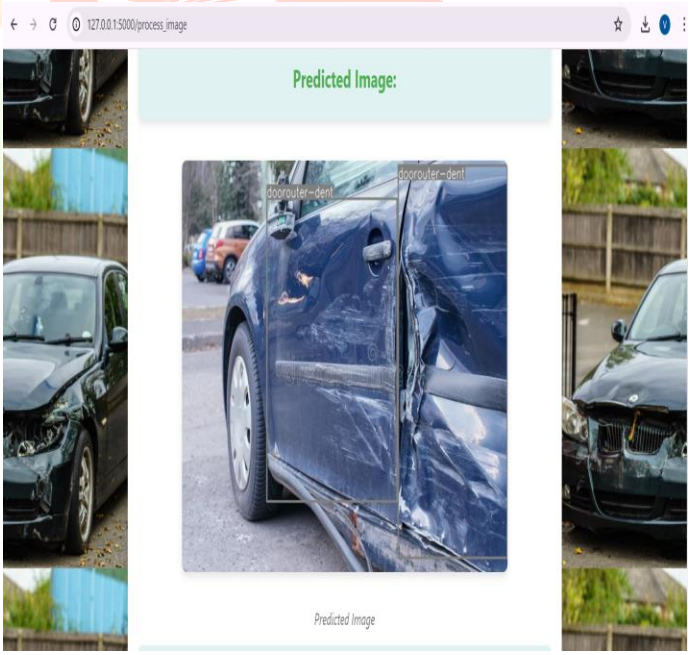

Fig – 6 IDENTIFYING THE DAMAGES

Fig – 7 BINARY MASK IMAGE

CONCLUSION

In conclusion,our paper presents a comprehensive approach for automated car damage detection using the YOLOv8 algorithm.We begin by collecting a diverse dataset of car images containing various types of damages,including dents, scratches,cracks,glass shatters,flat tires and broken lamps. Through image preprocessing and YOLOv8 algorithm training,we optimize the model's ability to accurately detect and localize these damages in real-time.The system categorizes images into damaged and undamaged classes and stores the trained model in a database for future use.During testing and analysis,the model efficiently processes test input images,generating binary maps that precisely highlight the location of damages.By masking exact damage locations,our system provides a clear visualization of the extent of damages, facilitating informed decision-making in car maintenance and insurance claim processing.Overall,our proposed system offers a practical and efficient solution for automating car damage assessment, contributing to enhanced vehicle safety, streamlined insurance claim processing and improved customer satisfaction.

REFERENCES

[1] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. 2019. A very deep transfer learning model for vehicle damage detection and localization. In2019 31st International Conference on Microelectronics (ICM). IEEE,158–161.

[2] Dorathi Jayaseeli, J. D., Jayaraj, G. K., Kanakarajan, M., & Malathi, D. (2021). Car Damage Detection and Cost Evaluation Using MASK R-CNN. In Intelligent Computing and Innovation on Data Science (pp. 279-288). Springer, Singapore.

[3] Q. Zhang, X. Chang and S. B. Bian, "Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN," in IEEE Access, vol. 8, pp. 6997-7004, 2020, doi: 10.1109/ACCESS.2020.2964055

[4] Kyu, Phyu Mar and Kuntpong Woraratpanya. "Car damage detection and classification." Proceedings of the 11th international conference on advances in information technology. 2020.