CAR DAMAGE DETECTION AND PRICE PREDICTION USING DEEP LEARNING

1Prof. Reshma Totare, 2Varad Bhalsing, 3Mayur Lende, 4Tejas Maramwar, 5Chirag Naikwadi
1Assistant Professor, 2Student, 3Student, 4Student, 5Student
1Department of Information Technology,
1AISSMS’s Institute of Information Technology,
Pune-1, Maharashtra, India

Abstract: This project proposes a solution for automating car damage detection and price prediction using deep learning techniques, namely convolutional neural networks (CNN) and transfer learning with VGG-16. The system accurately detects and classifies car damage while providing approximate repair costs based on severity and damaged location. It benefits insurance companies by streamlining claims assessment and individuals by providing independent claims assessment. Experimental evaluation demonstrates high accuracy and recall rates, making it a valuable tool for effective decision-making in automotive repair.

Index Terms - Car Damage Detection, Machine learning, Prediction, VGG16, Transfer Learning, Deep Learning, Price Prediction

I. INTRODUCTION

In today's society where the number of road accidents is on the rise, the insurance industry has recognized the importance of innovation, cutting-edge technology and artificial intelligence (AI) to stay ahead of the market. Insurance companies are investing heavily in solutions such as data analytics and processing, fraud detection, risk mitigation and claims automation. One area that poses a significant financial burden to insurance companies is accident avoidance, especially when it comes to vehicle damage estimates.

Artificial intelligence, specifically machine learning and deep learning, has emerged as a powerful tool in the insurance industry. These technologies offer promising solutions to accelerate claims analysis and facilitate faster claims settlement. However, developing effective applications for automotive damage assessment using deep learning remains a challenging task. Deep learning algorithms excel at solving complex problems, but require large data sets and significant computing resources to build models.

One of the significant obstacles in training deep learning models for car damage assessment is the lack of publicly available datasets containing photographs of damaged cars. Car damage assessment is a specialized field and obtaining sufficient labeled data for training purposes is challenging. Lack of data is a significant challenge as deep learning models thrive on massive amounts of data and provide robust performance.

In addition, the task of automatic damage detection and segmentation presents additional complexity. Existing approaches often suffer from reduced segmentation accuracy and lower detection speed. To address these challenges, this project leverages the power of VGG-16, a widely used pre-trained convolutional neural network (CNN) architecture, for transfer learning. Transfer learning allows us to use the learned VGG-16 features and fine-tune them for the specific task of vehicle damage detection and segmentation.

In this research, we propose a VGG-16-based model to detect and segment damaged regions of vehicles involved in accidents. The proposed model offers an efficient and accurate solution for the identification and limitation of vehicle damage and enables insurance companies to speed up the processing of insurance claims. Additionally, this model has the potential to benefit individual users trying to assess the severity of damage and predict repair costs.

Utilizing deep learning capabilities and transfer learning techniques, this project aims to provide a cost-effective and practical solution for vehicle damage detection and price prediction. The following sections of this paper will delve into the methodology, experimental results, and potential implications of this research in the context of insurance companies and other stakeholders involved in vehicle damage assessment.
Risk Analysis:

Risk analysis is a fundamental aspect to consider when implementing the proposed system that uses the VGG-16 algorithm to assess vehicle damage. There are several key risk factors to consider, including:

1. Image size: The performance of the VGG-16 algorithm can be affected by the size of the images used for training and testing. Large images may require more computing time and processing power, while small images may result in reduced accuracy. It is crucial to find the optimal image size that balances accuracy and computational efficiency.

2. Data Availability: The availability of high-quality and sufficient datasets is essential for training deep learning models, especially for specialized tasks such as car damage assessment. System performance strongly depends on the amount and quality of available data. Limited data sets can result in reduced accuracy, leading to false positives or false negatives.

3. Time management: The VGG-16 algorithm is computationally intensive and requires considerable computing power for efficient operation. Training the model and making predictions on new images can be time-consuming. Parallel processing techniques should be used to optimize algorithm parameters and reduce computation time.

4. Human intervention: Despite the high accuracy of the VGG-16 algorithm, there is always the possibility of errors or misclassification. Human intervention and validation should be incorporated into the vehicle damage assessment process to ensure the accuracy of the system output. Human experts can verify and correct any potential misclassifications made by the algorithm.

5. Privacy and security: The proposed system involves the use of sensitive data such as photos of car damage. Privacy and data security are critical factors. It is essential to comply with personal data protection regulations and to implement appropriate security measures to protect data from unauthorized access or breach.

By acknowledging these risks and implementing appropriate mitigation measures, the proposed system can be developed with increased reliability, accuracy, and security, ensuring its effective use for vehicle damage assessment using the VGG-16 algorithm.

II. LITERATURE REVIEW

A previous study [1] has used pre-trained deep learning models, MobileNet and YOLO, and have been applied on a customized vehicle damage dataset. YOLO algorithm is used for detection but YOLO’s full utilization is in classifying multiple objects. In this case, only a single object i.e. car is to be detected where VGG-16 can be used.

Another study by Umer Waqas, Nimra Akram, Soohwa Kim, Dongjun Lee, Jihoon Jeon [2] uses algorithm to classifies three types of vehicle damage categories which include medium damage, huge damage or no damage. Also, metadata analysis and moiré effect detection is used to verify the authenticity of the images uploaded by users.

In [3], an insight to further more improvement is suggested. Robust Mask RCNN algorithm is utilized which improves on previous work yet is still lacking in few areas. For example, the detection accuracy is very high, but the mask instance segmentation cannot be completely correct, and some areas in which the damage is not obvious cannot be segmented.

Another paper [4] proposes a deep learning-based solution for car damage classification. Since there was no publicly available dataset, they created a new dataset by collecting images from web and manually annotating them. Experimented with multiple deep learning-based techniques such as training CNNs from random initialization, Convolution Autoencoder based pre-training followed by supervised fine tuning and transfer learning.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture consists of different modules that work together to detect and analyse car damage from recorded images. The system is based on a web application that allows users to upload images and receive a report detailing the severity and estimated cost of the damage.
The modules are as follows:

**Web application:**
The system's user interface is built using HTML, CSS and Bootstrap and is integrated by Flask. Users can upload images of damaged cars or any random images that the algorithm needs to process.

**Module 1: Car detection:**
The first module of the proposed system uses the VGG-16 algorithm to detect whether the uploaded image contains a car or not. If the image does not contain a car, the algorithm returns an error message and the process stops.

**Module-2: Damage Detection:**
The second module uses three different algorithms: VGG-16, VGG-19 and ResNet-50 to detect car damage in the recorded image. The algorithm analyses the image and detects the damaged area.

**Module-3: Damage Side Detection:**
This module determines which side of the vehicle has sustained damage. Based on the analysis of the recorded image, the system identifies whether the damage is in the rear, front or side of the vehicle.

**Module-4: Damage Severity Detection:**
The severity of the damage is determined by analysing the damaged area using a heat map. A heat map is a two-dimensional visual representation of data where values are color-coded, providing a convenient and clear view of information.

**Module 5: Predicting Damage Costs:**
Based on the location and severity of the damage and the model of the car, the system predicts the estimated cost of repairing the damage. This module uses machine learning algorithms to accurately predict repair costs.

The architecture of the proposed system is designed to provide a comprehensive solution for car damage assessment, starting from car and damage detection to repair cost prediction. By combining different modules, the system can accurately detect the location and severity of damage, which allows insurance companies to handle insurance claims faster and more efficiently. The system's user-friendly interface makes it easy for users to upload images and receive an accurate report detailing damage and repair costs.

**IV. METHODOLOGY AND ALGORITHM**
System implementation involves the use of various libraries and packages to build and deploy a car damage detection and price prediction system. Some of the key components used in the implementation are:

- **Keras and TensorFlow:** Keras is a high-level neural network API that runs on top of TensorFlow, a popular deep learning framework. These libraries provide a wide range of features and tools for building and training deep neural networks, which are crucial for image classification and damage detection tasks.

- **VGG-16:** The VGG-16 model is a pre-trained deep convolutional neural network that has been shown to be effective in image classification tasks. It is widely used for its ability to extract meaningful features from images and has been applied to various computer vision tasks, including car damage detection.

  The VGG-16 architecture consists of 16 layers, including convolutional layers, max-pooling layers, and fully connected layers. It uses small receptive fields (3x3) for convolution operations and max-pooling layers for resampling the spatial dimensions of feature maps.

  The VGG-16 implementation involves loading a pre-trained model, feeding the car images into the model, and extracting features from the last convolutional layer. These extracted features are then used as input for subsequent classification or regression tasks, such as determining the presence of car damage or predicting damage severity.

  Leveraging the power of VGG-16, our system can effectively analyse and classify car images, enabling accurate damage detection and facilitating repair cost prediction. The use of VGG-16 increases the performance and reliability of our car damage detection system, provides valuable information for insurance companies and automates the inspection process.

- **Image processing libraries:** Various image processing libraries such as OpenCV, PIL (Python Imaging Library) or scikit-image are used for image manipulation, including resizing, cropping and enhancing input images for optimal analysis.

- **NumPy and Pandas:** NumPy and Pandas are essential libraries for data manipulation and analysis. They provide efficient and convenient methods for manipulating numerical data, processing arrays, and performing the operations necessary for data pre-processing and feature extraction.

- **Scikit-learn:** Scikit-learn is a popular machine learning library that offers a wide range of algorithms and tools for classification, regression and data pre-processing. It can be used for tasks such as feature selection, data partitioning, and model evaluation.

- **Flask:** Flask is a lightweight web framework in Python used to create a web application interface of a car damage detection system. It enables easy integration of machine learning models and facilitates interaction between the user and the system. These libraries and packages form the basis of the system implementation and enable efficient data processing, model training and prediction. They provide the necessary tools and functions to handle the complex tasks of image analysis, classification and price prediction in the vehicle damage detection system.
V. PROJECT DESIGN

The project design includes an activity diagram with swim lanes that illustrates the flow of activities and responsibilities between different components. Swim lanes represent the various modules and their interactions within the system.

1. Web Interface: This swim lane represents the user interface component that allows users to interact with the system. It includes activities such as taking and uploading images of the damaged car, starting the detection process and receiving the final results.
2. Car detection: In this swimming lane, activities related to car detection are performed. This module analyzes the uploaded image and identifies the presence of the car. It checks whether the image contains a car or not and proceeds accordingly.
3. Damage Detection: The damage detection swim lane focuses on identifying damaged areas of the car. This module processes the image to precisely locate and segment damaged areas. It uses advanced computer vision techniques and deep learning models to detect and highlight damaged areas.
4. Location and Severity: This swimming lane deals with activities related to determining the location and severity of detected damages. It analyzes the segmented areas and classifies them into pre-defined categories such as front, side or rear damage. In addition, it assesses the severity of damage and categorizes it as minor, moderate or major.
5. Classification: This swim lane links the location and severity of the damage found. Maps identified locations with corresponding severity levels. For example, it associates damage to the front with less severity or damage to the rear with more severity. This classification step ensures accurate categorization of damages.
6. Price Prediction: Swimming Price Prediction uses historical data and regression analysis to estimate repair costs based on detected damages. It considers factors such as car make, model, year and severity of damage to predict approximate repair costs. And finally, result is displayed.
VI. GUI AND EXPERIMENTAL RESULTS

User Interface has been developed.

The GUI is designed to be user-friendly and intuitive, with clear instructions and simple navigation menus. The use of HTML, CSS and Bootstrap ensures that the interface is responsive and adapts to different screen sizes and devices.

Figure 3 shows the home page of the proposed system, where image can be uploaded.

Figure 4 shows the uploaded image which can be submitted or changed accordingly.

Once the image is submitted, its result such as damage, damage location, damage severity and approximate price is displayed, which is depicted in figure 5.

Fig. 3: Interface to add image

Fig. 4: Image Uploaded
VII. RESULT AND EXPERIMENTS

The following section presents the results and experiments conducted for the proposed system, including the use of confusion matrices, precision, and recall tables. Confusion matrices are utilized to evaluate the performance of the system in detecting whether an image contains car damage or not, as well as determining the location of the damage (front, side, or rear).

For the damage detection, we got 91% Precision, 90% Recall as well as 90% F1-score.

As for the damage location, we got 75% Precision, 75% Recall as well as 75% F1-score.

- Confusion Matrix - Damage Detection:

Fig-6: Damage Detection Confusion Matrix

The above figure shows the confusion matrix of Damage Detection where 460 images are considered out of which 194 undamaged cars are detected as undamaged and 221 damaged cars are detected as damaged while 45 images are not identified correctly.
Table 1: The precision, recall, and f-measure for Damage Detection

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>f1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.96</td>
<td>0.84</td>
<td>0.90</td>
<td>230</td>
</tr>
<tr>
<td>1</td>
<td>0.86</td>
<td>0.96</td>
<td>0.91</td>
<td>230</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
<td>460</td>
</tr>
</tbody>
</table>

Table 1 shows the precision, recall, and f-measure for damage detection during the validation of the system.

- Confusion Matrix – Damage Location

![Damage Location Confusion Matrix](image)

The above figure shows the confusion matrix for Damage Location.

Table 2: The precision, recall, and f-measure for Damage Location

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>f1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.77</td>
<td>0.86</td>
<td>0.81</td>
<td>73</td>
</tr>
<tr>
<td>1</td>
<td>0.73</td>
<td>0.66</td>
<td>0.69</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>0.69</td>
<td>0.72</td>
<td>48</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>171</td>
</tr>
</tbody>
</table>

Table 2 shows the precision, recall, and f-measure for damage location during the validation of the system.

VIII. CONCLUSIONS

To deal with the compensating problem of damaged autos, the model proposed here employs a deep learning-based detection technique for vehicle-damage identification. The suggested approach of transfer learning-based damage detection of the vehicle is generic after testing, and can also better adapt to the diverse elements of damaged car images. Even though the model was trained on a very small dataset, good results were achieved. Data extension can be done in the future to raise the dataset's size, gather additional automobile damage images under various degrees of illumination and weather conditions, enrich the data, the edge-contour enhancement of images can be improved and the damaged parts of the car can be masked more accurately. Also, the model can be further enhanced to predict the repair price of the damaged area by extracting the predicted part details like the segmented mask area.
References


