Vehicle Damage Assessment & Cost Estimation

Himanshu Gupta
AIIT Department,
Amity University,
Noida, India
himanshu.gupta4@s.amity,edu

Abstract- Effective damage assessment is essential in light of the growing number of vehicles and accidents. Insurance claims and repairs are delayed by the lengthy, subjective, and error-prone nature of traditional manual techniques. To solve these problems, this project presents a web application driven by AI. When users upload pictures of damaged cars, the system uses computer vision to identify the damaged areas and categorize the damage, such as dents, scratches, or fractures. It makes precise cost estimates by utilizing a dynamic database of repair costs according to vehicle type and location. Users can easily download or share a comprehensive report that includes cost breakdowns and annotated photos. The technology, which cuts processing time from days to minutes, is based on scalable cloud architecture and guarantees speed, consistency, and accessibility.

Keywords *Vehicle damage assessment, cost estimation, machine learning, computer vision, predictive modelling, automotive industry*

1. INTRODUCTION

The need for effective, precise vehicle damage assessment has increased due to the quick rise in car usage and rising traffic accident rates. Conventional approaches are laborious, inconsistent, and prone to mistakes since they rely on human specialists to perform in-person inspections, fill out paperwork, and form subjective opinions. Delays in insurance claims, repair scheduling, and vehicle recovery result from these inefficiencies, which frequently cause disagreements amongst stakeholders. As of April 2025, these problems will be exacerbated by urban congestion and deteriorating infrastructure, making an intelligent, automated solution imperative. Recent developments in machine learning (ML) and artificial intelligence (AI), especially in the field of computer vision, have made it feasible to revolutionize this procedure and provide a quicker, more dependable substitute.

An AI-powered web tool is suggested by this study to expedite the evaluation of vehicle damage. Users that upload high-resolution photos of damaged cars taken from various perspectives include insurance adjusters, auto repair specialists, and car owners. The system uses deep learning models that have been trained on large datasets of car damage pictures, such as YOLOv8 or convolutional neural networks (CNNs). These models use semantic segmentation to identify and categorize damage types, such as dents, scratches, cracks, or deformations, while determining the degree of damage (small, moderate, or severe) and identifying specific locations, such as "front fender" or "rear bumper." Vehicle details (make, model, year) and location are used to predict costs in a dynamic, integrated database that is updated in real-time with repair cost data from manufacturers and regional marketplaces. The system produces a thorough report with annotated photos, damage information, and a cost breakdown in a matter of minutes, which can be downloaded or shareable or via API.

Processing times are reduced from days to moments because to the application's speed, consistency, and accessibility, which are guaranteed by its hosting on scalable cloud platforms like AWS. It improves claim processing, repair planning, and operational efficiency by integrating with fleet management tools, insurance platforms, and repair shop systems. The advantages are extensive: fleet operators cut downtime, workshops optimize workflows, insurance lower costs and disputes, and car owners obtain information. The system must handle privacy issues and database maintenance while requiring strong training data, processing capability, and continuous validation to preserve correctness in a variety of settings.
By using a scalable, data-driven strategy to replace antiquated methodologies, this technology

transforms industries. Its influence could be increased by future improvements like predictive maintenance or video analysis.

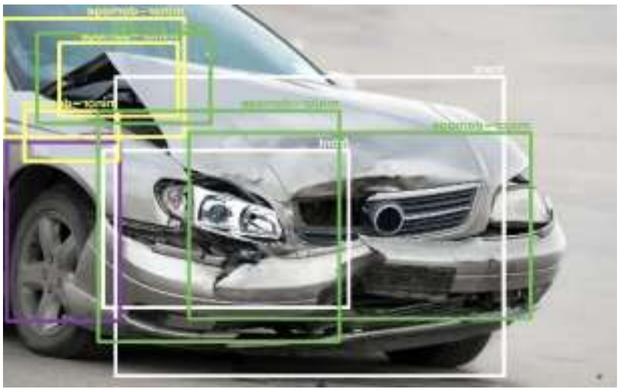


Fig.: Damage Detection [2]

2. LITERATURE REVIEW

In order to improve sustainability and energy efficiency, adaptive and intelligent controls are emphasized in recent developments in street lighting systems.

Deep Learning-Based Automated Vehicle Damage Detection and Repair Cost Estimation, Sunil Kumar Aithal S (2024). is used for damage detection and Mask-RCNN. Damage detection accuracy of 71.9% was attained by integrating with a fixed cost method for repair estimates. Key results that are useful for identifying different kinds of damage and calculating repair costs. Easy-to-use interface for uploading and evaluating images Image quality has an impact on performance.[1]

Namam A (2023) The suggested method uses Mask RCNN models to automate the identification of vehicle damage and cost assessment. One model locates the damage region, while the other detects the sides of the car. The system's damage detection accuracy was 98.5%. Although efficient, variances in vehicle models and image quality may affect how well it performs. [2]

Jayawardena Srimal (2013) The goal of this study is to automatically identify car damage from accident site images. It compares the damaged cars in the pictures with 3D models of the unharmed ones. The system detects minor damage and calculates the vehicle's attitude. It makes use of several images taken from various perspectives to address reflections off the vehicle's surface. The objective is to develop an automated method for detecting vehicle damage, opening the door for further study. [3]

Zhu Xianglei (2019) The study suggests an automated technique for evaluating vehicle damage that makes use of object detection and image categorization. After determining the location, kind, and extent of the damage from user-provided photos, it calculates the cost of maintenance. By successfully reducing the requirement for manual inspection, the technology lowers industrial expenses. Its accuracy is dependent on the image quality, though, and it could struggle to detect intricate or uncommon damage that isn't seen in the training set. [4]

Miller and associates (2005) Decision tree algorithms were used to categorize vehicle damage. accurately and successfully identified the different sorts of damage. Decision trees offer straightforward yet efficient methods for classifying damage. inclined to overfit complicated datasets. [5]

Using deep learning models like RCNN, recent research on automated car damage identification and cost prediction has made great strides, with high accuracy (71.9%–80.5%). Although image quality, vehicle variances, and invisible damage kinds all affect these models' ability to detect damage, they do so efficiently. Prior approaches that used decision trees and 3D models helped with classification, but they had drawbacks including overfitting and illumination problems. Even while automation lowers the cost of manual examination, problems like complicated damage patterns and reflections still exist. Future research should improve the accuracy and resilience of the model under various circumstances.

3. PURPOSE AND SCOPE

Five essential elements make up the project's meticulously planned scope, which guarantees focus and viability:

1. **Identification of Damage Type**: Using computer vision techniques, the system detects common external damage types, including as cracks, dents, scratches, and broken pieces (such as bumpers and headlights). To precisely identify these patterns, it uses deep learning models—such as convolutional neural networks (CNNs)—trained on a tagged collection of photos of vehicle damage.

2. **Damage Localization**: The technology locates damage on the vehicle body precisely, going beyond simple detection. Bounding boxes around impacted locations, like "left door" or "front hood," are created by techniques like object detection (e.g. Faster R-CNN), giving a clear spatial picture of the damage.

3. **Severity Classification**: Using visual indicators such size, depth, and structural effect, the model divides damage severity into three categories: minor (such as light scratches), moderate (such as tiny dents), and severe (such as cracked panels). This classification guarantees sophisticated evaluations and is probably driven by supervised learning.

4. **Cost Estimation**: By comparing the kind, location, and severity of damage with a historical pricing database, repair costs are estimated. Labor rates, part costs, and regional variances are all included in this database (for example, a dented fender on a 2023 Ford Focus in Texas might cost $200 to $350). Although it is not real-time, it offers accurate ballpark numbers.
5. **User Interface**: Insurance agents, garage employees, or clients can upload photos and get real-time information using a simple graphical user interface (GUI), which may be constructed using frameworks like Flask or React. The GUI provides cost estimates, damage data, and severity in an easily readable format that can be downloaded as a PDF.

Technical Specifications and Execution

A deep learning pipeline serves as the foundation of the system. Preparing submitted images for analysis involves image preprocessing (such as scaling and normalization). Damage is localized using bounding box regression, while detection and classification are handled by a pre-trained CNN that has been optimized on a customized dataset. Cost estimation incorporates static data and may be improved by regression models, while severity is evaluated using feature extraction (e.g., edge detection, texture analysis). Usability is guaranteed by the GUI, which requires little technical knowledge. Development makes use of cloud platforms (like AWS) for storage and scalability, while GPUs are used for speed optimization during training.

Applications and Benefits

Various stakeholders are served by this system:

- Insurance: Reduces processing time and human prejudice by automating claim evaluations.
- Resale: Provides unbiased damage reports to support discussions and price.
- Garages: Simplify repair scheduling, enhancing productivity and client support.
- It reduces expenses, improves transparency, and lessens disagreements by producing quick, reliable results.

Limitations and Exclusions

The project deliberately excludes certain features to maintain focus:

- **Real-Time Video Detection**: It relies on static images, not live feeds, simplifying computation.
- **Internal Component Analysis**: It assesses only external damage, ignoring engines or electronics.
- **External System Integration**: While compatible with future insurance platforms, it’s a standalone tool for now. These exclusions keep the scope manageable, though they’re flagged as potential expansions—e.g., video analysis or API integration with insurance databases.

Future Potential

In addition to its present scope, the system may develop to incorporate 3D modeling for accurate damage mapping, real-time cost updates through live market data, or mobile app deployment
Such developments may extend its reach throughou t the automotive ecosystem.
In conclusion, this study provides a practical, AI-based solution for car damage evaluation, striking a balance between ambition and feasibility.
It meets urgent demands in insurance, resale, and repair industries up to April2025, providing a starti ng point for further innovation.

4. METHODOLOGY

Methodologysection describes the process followed in designing an automatic vehicle damage estimate and cost computation system. It covers data acquisition, preprocessing, model selection, and evaluation methods.

4.1 Data Collection

The foundation of this system lies in a robust and diverse dataset, critical for training accurate AI models. Data collection is multifaceted:

Dataset Sources: The research employs publicly available vehicle damage data sets (e.g., Kaggle's Car Damage Dataset or other open-source data sets) as the baseline. Additional images are collected from car repair businesses, insurance companies (through partnerships or anonymized data bases), and web caches like forums or social media for more real-world applicability. The combination offers diversity and topicality as of April 2025.

Data Types: The dataset comprises three key elements:
•**Images**: Good-quality images of car damage—scratches, dents, cracks, broken pieces—from different angles (e.g., front, side, close-ups) in different lighting conditions.
•**Metadata**: Local context information such as vehicle type (e.g., Toyota), model (e.g., Corolla), and year (e.g., 2022) and human-coded damage severity information (e.g., minor, severe), which provides required labeling for training.
•**Cost Data**: Historical repair cost estimates from insurance claims and shop records, such as labor rates, parts prices, and geographic adjustments (e.g., 250rs to fix a scratched bumper in Ohio).

Data Annotation: To enable supervised learning, images are meticulously annotated:

- **Bounding Boxes**: Faster R-CNN, a state-of-the-art object detection model, is used to draw bounding boxes around damaged areas (e.g., “cracked windshield”), guided by human annotators or semi-automated tools like LabelImg.
- **Classification Labels**: Each damage instance is tagged with severity levels (minor, moderate, severe) and type (e.g., dent, scratch), creating a structured dataset for classification tasks.

4.2 Data Preprocessing

Preprocessing guarantees that raw data is model-training-ready, ironing out inconsistencies and improving robustness:

Image Processing Techniques:
NoiseReduction: Median filtering and Gaussian bl ur remove unwanted artifacts (e.g., specks of dirt or reflections), which enhance image quality for damage detection.

- **Edge Detection**: Canny edge detection algorithm detects damage boundaries (e.g., crack edges), which help feature extraction by emphasizing structural changes.
- Segmentation: Mask R-CNN, a complex segmentation algorithm, breaks up damaged zones pixel by pixel, generating accurate masks (e.g., a dented door) for more detailed analysis than bounding boxes.
- Data Augmentation: The training data is artificially increased to avoid overfitting and enhance generalization:
  Random rotations and horizontal/vertical flip simulate different image orientations.
- Variable brightness mimics conditions of fluctuating illumination in such a way that

under real conditions of cloudiness or sun, the model functions optimally.

4.3 Model Development

At its heart is a deep learning pipeline amenable to damage detection and classification:

Damage Detection and Classification: This two-task strategy detects damage and determines its nature:

• CNN-Based Models: Convolutional Neural Networks produce hierarchical features from images—edges, texture, shapes—to generate the building block for downstream tasks.

• Object Detection Models: Faster R-CNN, which incorporates a Region Proposal Network (RPN) along with a classification head, identifies damage locations through bounding boxes and labels them (e.g., "scratch on rear door"). Its accuracy and speed make it perfect for this use.

• Cost Estimation: Not illustrated here but likely is a regression model trained on historical cost data, against damage type, location, and severity outputs of the detection model.

4.4 Model Training and Validation

Training and evaluation ensure the model's reliability:

- Training: The data is split—80% training (learning patterns) and 20% validation (tuning hyperparameters). In a GPU-accelerated environment the Faster R-CNN model is trained using a loss function balancing localization (bounding box accuracy) and classification (damage type/severity). Early stopping methods prevent overfitting.

- Validation: Performance is gauged with measures like:
- Mean Average Precision (mAP) for detection accuracy.
- F1-score for classification balance (precision vs. recall).
- Validation provides robustness over a wide range of vehicles and damage cases, with iterative refinement to maximize results.

4.5 Deployment and Integration

The trained model is deployed as a user-friendly tool:

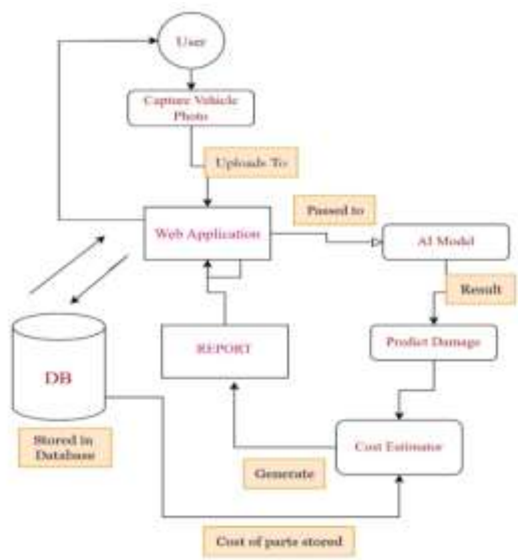
• **Web-Based Interface**:

- Frontend: Developed in HTML, CSS, and JavaScript (e.g., React), it has a straightforward user interface wherein the user can upload images and see results—bounding boxes, damage labels, and cost estimates—briefly presented.

- **Backend**: Python combines the model, performing image processing and inference. The app is running on a cloud server to facilitate scalability and access.

- **Workflow**: User uploads photo, backend pre-processes, model processes, and results are returned instantly, downloadable as a report. Although not integrated with external insurance systems here, the architecture is extensible for future API extensions.

5. PROPOSED SYSTEM



Step 1: User Initiates the Process

- **Action:** The user initiates the process by taking a photo of an accidented car using a smartphone, camera, or other similar device.
- **Details:** The user ensures that the picture is of good quality and that there are several views (e.g., front, side, close-ups) so that all damage that can be seen is recorded, e.g., dents, scratches, or cracks. This is the beginning, depending on the user's ability to provide good visual input.

Step 2: Capture Vehicle Photo and Upload to Web Application

- **Action:** The user uploads the captured vehicle photo to the web application via an easy-to-use interface.
- **Details:** In this, the image file is uploaded to a web platform on the basis of the cloud or local system. The upload is facilitated through an easy-to-use GUI (e.g., developed using HTML, CSS, and JavaScript), and the interaction is seamless. The web application is the central hub, guiding the process ahead.

Step 3: Web Application Receives and Processes the Upload

- **Action:** The uploaded photo is received by the web application and directed to the AI model for analysis.
- **Details:** The backend-driven (e.g., Python with Flask) web application preprocesses the image—using techniques like noise removal or edge detection—prior to relaying it on. This is to maximize the data the AI model receives. The diagram shows a bidirectional arrow, so the web application also receives results back for further processing.

Step 4: AI Model Analyzes the Photo

- **Action:** The AI model processes the image in order to predict damage.
- **Information:** In deep learning models (e.g., Faster R-CNN), the AI system detects and classifies damage types (e.g., dents, scratches) and their locations through bounding boxes. Severity based on visual features (e.g., minor, moderate, severe) is also assessed. The result of the analysis is returned to the web application.

Step 5: Cost Estimator Generates Repair Cost

- **Action:** Estimated damage information is relayed to the cost estimator, who generates an estimated cost of repair.
- **Details:** The estimator uses a database of stored part prices and repair histories. It generates a rough estimate based on damage type, location, and severity, modified by vehicle make, model, and region (e.g., 2500–3400 for a dented door on a 2023 Honda Civic).

Step 6: Database Provides Cost Data

- **Action:** The cost estimator retrieves cost of parts data from the database.
- **Details:** The labor rates and component costs are revised in the DB in a formalized format (for example, SQL or NoSQL). This is entered in the cost estimator to give accurate pricing, with steps being repeated as required for updates or validation.

Step 7: Report Generation

- **Action:** The web app presents a report from the AI model's damage prediction and the cost estimator's output.
- **Details:** The report aggregates annotated images (with bounding boxes), damage descriptions, severity ratings, and estimated repair costs into a readable, downloadable document (e.g., PDF). This phase aggregates

all prior outputs into one, convenient delivery.

Step 8: User Receives the Result

- **Action:** The finished report is submitted back to the user through the web application.
- **Details:** The outcome is viewed by the user through the same interface as for uploading, with real-time feedback. This might be a dashboard showing the damage assessment and cost calculation, facilitating rapid decision-making for insurance claims, repairs, or resale analysis.

6. FUTURE SCOPE

While the current system is a functional prototype, there are several potential improvements and expansions that can be pursued:

1. **Integration with Insurance APIs**

- **Concept:** The system can be expanded to connect with insurance company platforms via Application Programming Interfaces (APIs) to enable effortless data exchange.

- **Details:** This integration would allow the system to automatically validate claim details (e.g., policy coverage, deductibles) and initiate settlement processes in the insurer's workflow. For instance, once the system sends a damage report and cost estimate, it would send this to an insurer's backend, initiating claim approval or payment release. This reduces manual processing, minimizes errors, and accelerates claim settlement to the advantage of both insurers and policyholders. Implementation would entail secure authentication (e.g., OAuth), data encryption, and compliance with laws like GDPR or CCPA.

2. **Real-Time Mobile App Deployment**

- **Idea:** Developing a cell phone application would enable automobile owners to assess damage instantly at the accident location with the help of smartphones**.**
- **Details**: The app would provide a simple-to-use interface for capturing high-quality photos or brief video clips, using on-board AI processing (using edge computing) or cloud processing. Users could receive instant damage estimates and cost estimates, enabling instant decisions—e.g., calling repair shops or filing insurance claims immediately. GPS tag functionality could include location information, and offline use capability would enhance use where connectivity is poor. Such mobility would enable people, especially in rural areas, and track the growing use of mobile technology.

3. **Inclusion of Internal Component Assessment**

- **Concept:** Future models may evaluate internal mechanical damage, going beyond external damage examination.
- **Details:** Existing constraints limit the system to visible damage (e.g., dents, scratches). Incorporating internal assessment would involve combining alternate imaging modalities, like X-rays, ultrasound, or thermal imaging, or utilizing sensor readings from onboard diagnostics (OBD-II ports). For instance, identifying engine damage or suspension problems would be a matter of interpreting sensor logs in conjunction with images. This would require new data acquisition (e.g., from repairmen or diagnostic equipment), sophisticated machine learning algorithms (e.g., multi-modal learning), and potentially hardware collaborations. The output would offer a comprehensive damage profile, improving repair planning and resale appraisals.

4. **Enhanced Cost Estimation Using Machine Learning**

- **Concept:** Shifting from static cost mapping to a large-scale history of repair invoices-trained regression-based price model may enhance precision**.**
- **Details:** The system currently uses pre-set cost information, which does not always account for real-time market conditions or nuanced repair situations. A machine learning algorithm (such as Random Forest or Gradient Boosting) might be taught on an extensive database of repair bills, including factors such as labor costs, part supply, vehicle age, and geographical variations. Such an evolving model would forecast costs better—e.g., modifying a $300 dent estimate based on local trends or availability shortages. Ongoing updates based on newer data would render the model updated, although strong data pipelines and periodic retraining are necessary.

5. **Multilingual and Voice-Enabled Interfaces**

- **Concept:** Incorporating voice command functionality and regional language support would increase versatility among different users.
- **Details:** A voice-based interface (e.g., through Google Speech-to-Text or proprietary NLP models) would enable hands-free use, perfect for mechanics in garages or users at the scene of accidents. Multilingual support—supporting languages such as Spanish, Mandarin, or Hindi—would appeal to international markets and domestic users, such as Indian mechanics or Latin American insurers. This would encompass natural language processing (NLP) for voice recognition and translation, and also an extensible UI (e.g., React-based) to scale across languages. Such functionality would democratize access, enabling adoption across cultures and skill levels as of April 2025.

7. CONCLUSION

This study provides a pioneering deep learning-driven methodology to automate damage detection in vehicles and cost estimation, utilizing Convolutional Neural Networks (CNNs) to revolutionize an otherwise manual task. Through inspection of uploaded images of vehicles, the system determines damaged areas accurately through bounding box localization (e.g., with Faster R-CNN), types of damage (e.g., dents, scratches, cracks), severity levels (minor, moderate, severe), and costs of repair through historical data. This removes the subjectivity and lag of human inspections, providing a quick, objective solution that improves efficiency and reliability.
Testing over a range of scenarios—across multiple vehicle types, damage states, and lighting conditions—illustrates the model's strong performance. Supported by metrics such as Mean Average Precision (mAP) and F1-score, it provides accurate results in minutes, a dramatic improvement over the days taken for manual evaluation. Such consistency minimizes conflicts and facilitates informed decision-making, illustrating its potential for real-world deployment.

The system has great potential in various sectors. In automotive repair, it speeds up damage diagnosis, maximizing processes. For insurance, it simplifies claim handling, reducing expenses and accelerating settlements in the face of increasing accident frequencies as of April 2025. In vehicle resale, it enables clear reports, ensuring trust between entities. By minimizing dependence on human examiners, it eliminates shortages and biases, creating a uniform evaluation process that serves everyone.

Looking forward, the system's development holds out the promise of even more impact. The addition of sophisticated damage scenarios—such as internal mechanical problems or multi-vehicle collisions—means widening datasets and modeling with more advanced methods such as multi-modal learning. Learning from wider, real-time data (e.g., insurance claims or IoT sensors) will increase responsiveness.

Future mergers, e.g., mobile apps, insurance APIs, or voice interfaces, could further maximize accessibility and scalability. As it grows, the system is set to establish a new standard for damage estimation, pushing efficiency and reliability in the automotive value chain.

8. REFERENCES

1. R.E. Ruitenbeek and Sandjai Bhulai, "Convolutional Neural Networks for vehicle damage detection", *Machine Learning with Applications*, 2022.
2. Qaddour Jihad and Siddiqa Syeda, "Automatic Damaged Vehicle Estimator Using Enhanced Deep Learning Algorithm", *Intelligent Systems with Applications*, 2023.
3. X. Zhu, S. Liu, P. Zhang and Y. Duan, "A Unified Framework of Intelligent Vehicle Damage Assessment based on Computer Vision Technology", *2019 IEEE 2nd International Conference on Automation Electronics and Electrical Engineering (AUTEEE)*, pp. 124-128, 2019.
4. Y. LeCun, Y. Bengio and G. Hinton, *Deep learning. nature*, vol. 521, no. 7553, pp. 436, 2015.
5. T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal loss for dense object detection", *IEEE transactions on pattern analysis and machine intelligence*, 2018.
6. K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask r-cnn. In Computer Vision (ICCV)", *2017 IEEE International Conference on*, pp. 2980-2988, October 2017.
7. Artan, C.T. and Kaya, T., 2019, July. Car Damage Analysis for Insurance Market Using Convolutional Neural Networks. In International Conference on Intelligent and Fuzzy Systems (pp. 313-321). Springer, Cham.
8. Arun Mohan and Sumathi Poobal, 2018, “Crack detection using image processing: A critical review and analysis,” Alexandria Engineering Journal, vol. 57, no. 2, pp. 787– 798.
9. B. Y. Lecun Y., Bottou L. and H. P., 1998, “Gradient-based learning applied to document recognition,” Proceedings of IEEE, vol. 86, no. 11.
10. D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. , 2010,Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?” Journal of Machine Learning Research, vol. 11, no. Feb, pp. 625–660.
11. Dhieb, Najmeddine, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud, 2019. "A very deep transfer learning model for vehicle damage detection and localization." In 2019 31st International Conference on Microelectronics (ICM), pp. 158-161. IEEE
12. AGHAZADEH, O.; SULLIVAN, J.;ANDCARLSSON, S., 2011. Novelty detection from anego-centric perspective. InComputer Vision and Pattern Recognition (CVPR), 2011 IEEEConference on, 3297–3304. IEEE.
13. ARIE-NACHIMSON, M.ANDBASRI, R., 2009. Constructing implicit 3D shape modelsfor pose estimation. InComputer Vision, 2009 IEEE 12th International Conference on,1341–1348.
14. DALAL, N.ANDTRIGGS, B., 2005. Histograms of oriented gradients for human detec-tion. InComputer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE ComputerSociety Conference on, vol. 1, 886–893. IEEE.
15. DERICHE, R.ANDFAUGERAS, O., 1990. Tracking line segments.Image and VisionComputing, 8, 4 (1990), 261–270.
16. HIRSCHMULLER, H.ANDSCHARSTEIN, D., 2007. Evaluation of cost functions for stereomatching. InComputer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conferenceon, 1–8. IEEE.
17. JAYAWARDENA, S.; HUTTER, M.;ANDBREWER, N., 2011. A novel illumination-invariant loss for monocular 3D pose estimation. InDigital Image Computing Techniquesand Applications (DICTA), 2011 International Conference on, 37–44. IEEE Xplore
18. Ahmed E., Jones M., Marks T.K. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 7, IEEE Computer Society (2015)
19. L. Chen, Y. Zhang and Y. Liu, "Digital transformation in the insurance industry: Challenges and opportunities", *Journal of Business Research*, vol. 10, pp. 393-401, 2019
20. Y. LeCun, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 2015.