Intelligent Vehicle Damage Assessment &

Cost Estimator for Insurance Companies

DATE	**12/11/2022**
TEAM ID	**PNT2022TMID25973**
PROJECT NAME	**Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies**

Objective

By training convolutional neural networks, you can accurately classify vehicle damage using computer vision and deep learning techniques.

Use Case

- The rapidly expanding automobile industry strongly supports the equally rapidly expanding auto insurance market.
- Although, until now, this industry has relied solely on traditional methods of making repair claims.
- In the event of an unfortunate accident, claims for car damage must be filed manually.
- An inspector must physically inspect the vehicles in order to assess the damage and obtain a cost estimate.
- There is also the possibility of inaccurate settlements due to human error in such a situation.

- Automating such a process with the help of machine learning and remote usage would make the process much more convenient for both sides of the damage, increasing insurance carrier productivity and customer satisfaction.

- While the technology is yet to achieve the highest possible levels of accuracy, above is a proof of concept of the application of Deep Learning and Computer Vision into automating the damage assessments by building and training Convolution Neural Networks.

Solution

- The simplest way to automate such a system would be to create a Convolution Neural Network model capable of accepting images from the user and determining the location and severity of the damage.

- The model must pass a series of tests to ensure that the image is, in fact, of a car, and that it is damaged.

- These are the preliminary checks performed before the analysis begins. The damage check will begin once all of the gate checks have been validated.

- The model will predict the location of the damage as front, side, or rear, as well as its severity as minor, moderate, or severe.
- These are the preliminary checks performed before the analysis begins. The damage check will begin once all of the gate checks have been validated. The model will predict the location of the damage as front, side, or rear, as well as its severity as minor, moderate, or severe.

The model accepts an input image from the user and processes it across 4 stages:

1. **Validates that the given image is of a car.**

2. **Validates that the car is damaged.**

3. **Finds location of damage as front, rear, or side**

4. **Determines severity of damage as minor, moderate, or severe**

The model can also further be improved to:

1. **Obtain a cost estimate**

2. **Send assessment to insurance carrier**

3. **Print documentation**

Challenges

1. **The field of Computer Vision is yet developing and not mature enough to deal with modular phone camera quality images. Angle, lighting, resolution are factors that can easily cause major disruptions in image classification.**

2. **Car insurance settlement claims require near perfect accuracy to ensure the customer is not frauded in the process. Such models would be required to be trained on humongous datasets which are highly difficult to procure.**

3. **To run such heavy datasets to ensure maximum accuracy would be imposed by hardware restriction. Storing, training, and deploying such heavy datasets over the cloud would require expensive architecture.**

4. **While the computer can avoid human errors, there are often situation that would require such a model to flag for human assistance.**

5. **Systems running on the Cloud, especially those dealing monetary data are also heavily susceptible to cyber risks and require heavily structured frameworks to ensure customer data security.**

6. **Such a process will require a certain level of manual control and filter to avoid flooding of fraudulent insurance claims**.

Model Architecture and Pipeline

Our system architecture is built around the following modules:

1. **User Input: User submits an image containing the damage.**

2. **Gate 1: Checks to ensure the submitted image contains a car.**

3. **Gate 2: Checks to ensure the submitted image of the car is damaged avoiding fraudulent claims.**

4. **Location Assessment: Tests image against the pre-trained model to locate damage**

5. **Severity Assessment: Tests image against pre-trained models to determine the severity of damage.**

6. **Results: The results are sent back to the user and third party**

Tools and Frameworks Used

- Data Set Collection:

1. Google Images – data source

2. Stanford Car Image Dataset – data source

3. Import.io – online web data scraper

- Model Development:

1. TensorFlow – Deep Learning Library

2. Keras – Deep Learning Library

3. NumPy – Scientific numerical calculations library

4. Scikit-learn – Machine learning algorithms tools

- Web Development:

1. Flask – Python web framework

2. Bootstrap – HTML, CSS, JavaScript framework

- Development Environment:

1. PyCharm IDE – Python program development environment
2. Jupyter Notebooks – web application for interactive data science and scientific computing
3. Anaconda Virtual Environments – python virtual environment application

- Libraries Used:

1. numpy
2. pandas
3. matplotlib
4. sklearn
5. seaborn
6. pickle
7. IPython
8. collections
9. h5py
10. json
11. Urllib

Improving The Model

1. **The data set used in this application consisted of around 1500 images for the first gate check, while the classification models were trained on only 400 images per class,**

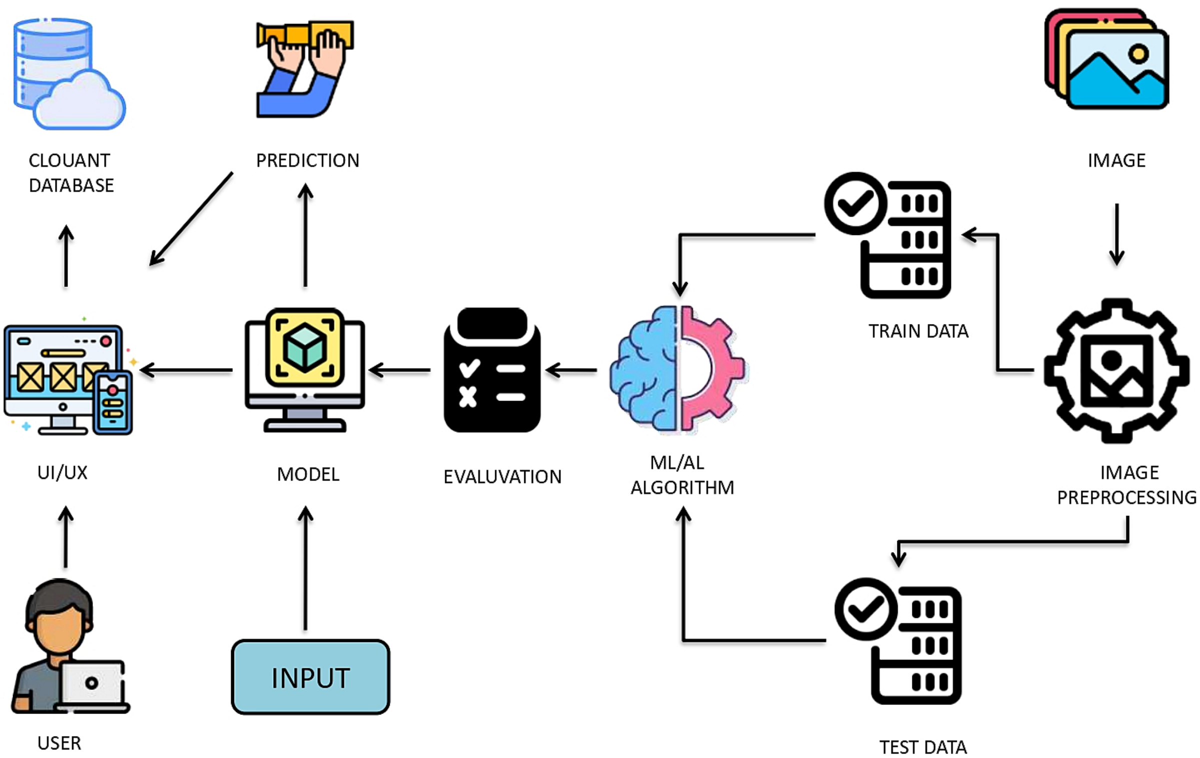
   while the validation dataset had approximately 75 to 100 images each class. Such a model will have low accuracy.

2. **With a wider range of data sets featuring multiple components of the car, the model can also be trained to identify what components are damaged, also classifying the varying degree of damage of each.**

3. **With a highly expansive dataset containing the make, model, year of the car and the possible cost estimates for the varying degrees of damage, the model can also predict the value for the user, before he submits the more advanced and detailed assessment for evaluation.**

4. **Using more secure and durable hardware, the entire system can be built on the Cloud to run remotely and from the user's cellular device itself.**

5. **The application can also be updated to recommend the use of policies pertaining to the specific accounts and other insurance benefits.**



CLOUANT DATABASE
PREDICTION
IMAGE
TRAIN DATA
UI/UX
MODEL
EVALUVATION
ML/AL ALGORITHM
IMAGE PREPROCESSING
USER
INPUT
TEST DATA