

Intelligent Vehicle Damage Assessment & Cost Estimator For Insurance Companies

Category: Artificial Intelligence

TEAM ID	PNT2022TMID21005
TEAM MEMBERS	<ul style="list-style-type: none">● SHYAMALAVANNAN M● DINESH J● SUDHARSAN S● SAI BARATH RAJ Y

Skills Required:

Python,CNN,IBM Cloud,IBM Watson,IBM Cloudant DB,Deep Learning,Python-Flask

Project Description:

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims.

The aim of this project is to build a VGG16 model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pics and the model can assess damage(be it dent from scratch from and estimate the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

Prerequisites

To complete this project, you must require the following software, concepts, and packages:

Anaconda Navigator:

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using a Jupyter notebook and Spyder.

To build Machine learning models you must require the following packages

- **Numpy:**
 - It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations
- **Scikit-learn:**
 - It is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy
- **Flask:**

Web framework used for building Web applications
- **Python packages:**
 - open anaconda prompt as administrator
 - Type “**pip install numpy**” and click enter.
 - Type “**pip install pandas**”and click enter.
 - Type “**pip install scikit-learn**” and click enter.
 - Type“**pip install tensorflow==2.3.2**” and click enter.
 - Type “**pip install keras==2.3.1**”and click enter.
 - Type “**pip install Flask**” and click enter.

- **Deep Learning Concepts**

- **VGG16:** VGG16 is a transfer learning method. A pre-trained model trained on 1000 classes of images.

VGG basic

- **Flask:** Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

Click here to learn more about flask : **Flask Basics**

Project Objectives

By the end of this project you will:

- Know fundamental concepts and techniques of VGG16.
- Gain a broad understanding of image data.
- Know how to pre-process/clean the data using different data preprocessing techniques.
- know how to build a web application using the Flask framework.

Project Flow:

- The user interacts with the UI (User Interface) to choose the image.
- The chosen image is analyzed by the model which is integrated with the flask application.
- VGG16 Model analyzes the image, then the prediction is showcased on the Flask UI.

To accomplish this, we have to complete all the activities and tasks listed below

- **Data Collection.**
 - Create Train and Test Folders.
- **Image Preprocessing.**
 - Import the ImageDataGenerator library
 - Configure ImageDataGenerator class
 - Apply ImageDataGenerator functionality to Train Set and Test Set
- **Model Building**

- Import the model building Libraries
- Loading the model
- Adding Flatten layers
- Adding Output Layer
- Creating Model Object
- Configure the Learning Process
- Train the Model
- Save the Model
- Test The Model
- **Cloudant DB**
 - Register & Login to IBM Cloud
 - Create Service Instance
 - Creating Service Credentials
 - Launch Cloudant DB
 - Create Database
- **Application Building**
 - Building HTML Pages
 - Build Python Code
 - Run The Application