Intelligent Vehicle Damage Assessment & Cost Estimator For Insurance Companies

IBM-Project-45850-1660732782

NALAIYA THIRAN PROJECT BASED ON LEARNING PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

Project Report

TEAM ID: PNT2022TMID47870

TEAM MEMBERS:

1. RAMESH R - 911719106022
2. DHANASEKARAN B – 911719106010
3. ANBURAJ S - 911710106003
4. MADHAVAN K – 911719106302

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

MOUNT ZION COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY, CHENNAI – 600 025

INDEX

13. APPENDIX

1. SOURCE CODE
2. GitHub & Project Demo Link

1. INTRODUCTION

Nowadays , a lot of money 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.

1.1 Project Aim

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 scratch from and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

1.2 Project Objectives

In today's world, it can observed that the number of vehicles we use is quickly expanding; let's agree that there isn't a single street without a car. As a result, an increase in the number of automobiles on the road may lead to an increase in the percentage of accidents occurring nearby; additionally, the number of accidents occurring nearby would be significant; the accidents would not be particularly serious, but the automobile would be damaged, prompting people to file insurance claims

1.3 Project Purpose

By reducing loss adjustment costs, improvements in the First Notice of Loss and the speed with which claims are examined and evaluated might save a lot of money in the automobile insurance claims process. Car damage is automatically identified and classified using advanced picture analysis and pattern recognition technology. A technique that compares before-and after-accident car images to automatically detect .

2. LITERATURE SURVEY

Paper 1:

K Kouchi and F Yamazaki, “Damage detection based on objectbased segmentation and classification from high resolution satellite can be studied. A strong earthquake of magnitude 6.8 struck the Mediterranean coast of Algeria on 21 May 2003 and the city of Zemmouri in Boumerdes province was most heavily damaged. QuickBird satellite observed the Zemmouri area on 23 May 2003. By image sharpening, buildings, cars and even debris can clearly be identified in a natural colour image. Preliminarily, the present authors performed visual damage inspection comparing the post-event image with an

image acquired before the earthquake. As a result, totally collapsed buildings, partially collapsed buildings, and buildings surrounded by debris were visually identified. Additionally, debris surrounding damaged buildings was also extracted. Although these observations indicate that high-resolution satellite images would be able to provide quite useful information to emergency management after natural disasters, it can also be said that the visual damage interpretation is time-consuming. For practical purposes, it must be necessary to complete damage detection as quick as possible after the occurrence of disasters in order to make use of the detection result in emergency management. Hence, an automated damage detection method, in which debris is identified, is

required to be developed. In this study object-based image segmentation and classification technique as well as pixel-based technique have been applied. This technique would make it possible to consider not only the spectral characteristics of objects but also the spatial relationship between objects that consist of homogenous pixels. For the purpose of investigating their effectiveness on identifying debris, the accuracy of the detection result has been assessed and compared with that of the pixel-based damage detection result.

Paper 2:

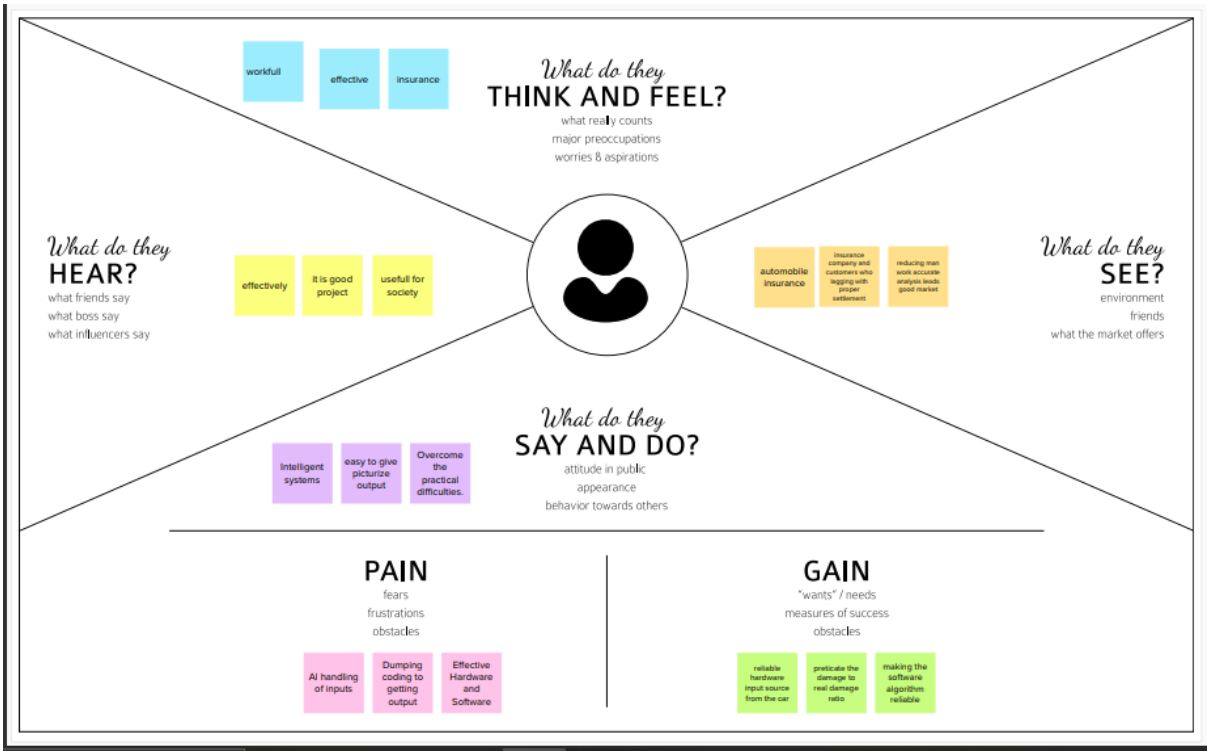
Michael Giering Mark R. Gurvich Soumalya Sarkar, Kishore K. Reddy,proposed “Deep learning for structural health monitoring: A damage characterization application,” in Annual Conference of the Prognostics and Health Management Society. Abstract and Figures Structural health monitoring (SHM) is usually focused on a fact of damage detection itself (e.g., Yes/No) or approximate estimation of damage size. Any additional details of the damage such as configuration, shape, networking, geometrical statistics, etc., are often either ignored or significantly simplified during SHM characterization. These details, however, can be extremely important for understanding of damage severity and estimations of follow-up damage growth risk. To avoid expensive human participation and/or over-conservative SHM decisions, solutions of computational recognition for damage characterized are needed. Autonomous SHM from visual data is one of the significant challenges in the field of structural prognostics and health monitoring (PHM). The main shortcomings of the image-based PHM algorithms arise from the lack of robustness and fidelity to handle the variability of environment and nature of damage types. In recent times, deep learning has drawn huge amount traction in the field of machine learning and visual pattern recognition due to its superior performance compared to the state-of-the-art techniques. The paper proposes to formulate and apply a deep learning technique to characterize the damage in the form of cracks on a composite material. The deep learning architecture is constructed by multi-layer neural network that is based on the fundamentals of unsupervised representational learning theory. The robustness and the accuracy of the approach is validated on an extensive set of real image data collected via applying variable load conditions on the structure. The paper has shown a high characterization accuracy over a wide range of loading conditions with limited number labeled training image data.

Paper 3:

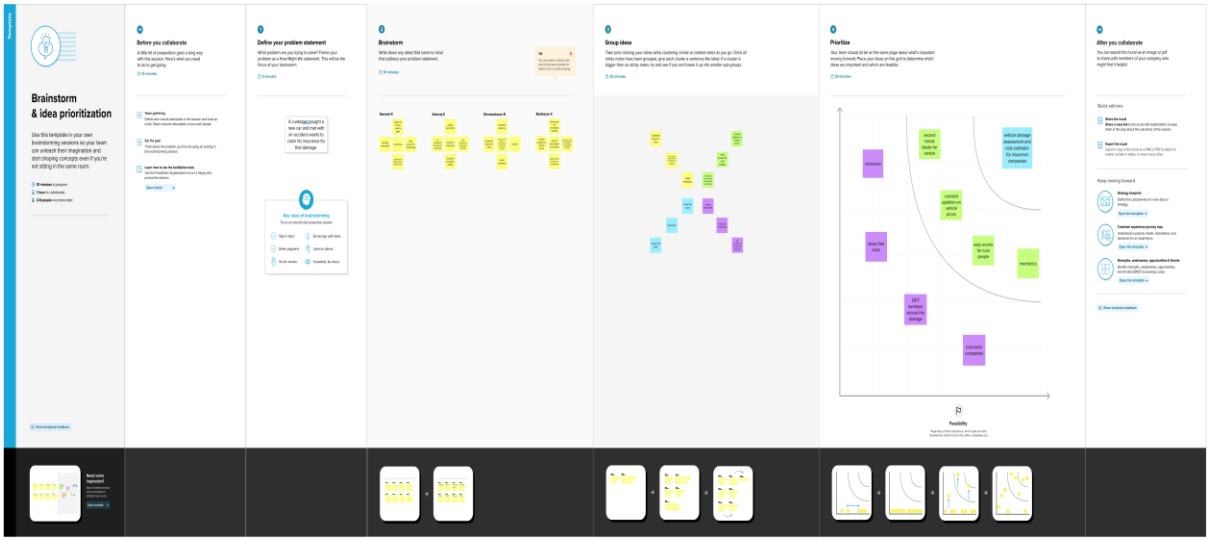
Maeda, Hiroya, proposed the paper "Road damage detection using deep neural networks with images captured through a smartphone. Research on damage detection of road surfaces using image processing techniques has been actively conducted, achieving considerably high detection accuracies. Many studies only focus on the detection of the presence or absence of damage. However, in a real-world scenario, when the road managers from a governing body need to repair such damage, they need to clearly understand the type of damage in order to take effective action. In addition, in many of these previous studies, the researchers acquire their own data using different methods. Hence, there is no uniform road damage dataset available openly, leading to the absence of a benchmark for road damage detection. This study makes three contributions to address these issues. First, to the best of our knowledge, for the first time, a large-scale road damage dataset is prepared. This dataset is composed of 9,053 road damage images captured with a smartphone installed on a car, with 15,435 instances of road surface damage included in these road images. In order to generate this dataset, we cooperated with 7 municipalities in Japan and acquired road images for more than 40 hours. These images were captured in a wide variety of weather and illuminance conditions. In each image, we annotated the bounding box representing the location and type of damage. Next, we used a state-of-the-art object detection method using convolutional neural networks to train the damage detection model with our dataset, and compared the accuracy and runtime speed on both, using a GPU server and a smartphone. Finally, we demonstrate that the type of damage can be classified into eight types with high accuracy by applying the proposed object detection method. The road damage dataset, our experimental results, and the developed smartphone application used in this study are publicly available Paper 4:]U. Waqas, N. Akram, S. Kim, D. Lee and J. Jeon, "Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning," 2020 IEEE Canadian Conference on Electrical and Computer Engineering. Image-based vehicle insurance processing and loan management has large scope for automation in automotive industry. In this paper we consider the problem of car damage classification, where categories include medium damage, huge damage and no damage. Based on deep learning techniques, MobileNet model is proposed with transfer learning for classification. Image-based vehicle insurance processing and loan management has large scope for automation in automotive industry. In this paper we consider the problem of car damage classification, where categories include medium damage, huge damage and no damage. Based on deep learning techniques, MobileNet model is proposed with transfer learning for classification. Moreover, moving towards automation also comes with diverse hurdles; users can upload fake images like screenshots or taking pictures from computer screens, etc. To tackle this problem a hybrid approach is proposed to provide only authentic images to algorithm for damage classification as input. In this regard, moiré effect detection and metadata analysis is performed to detect fraudulent images. For damage classification 95% and for moiré effect detection 99% accuracy is achieved.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



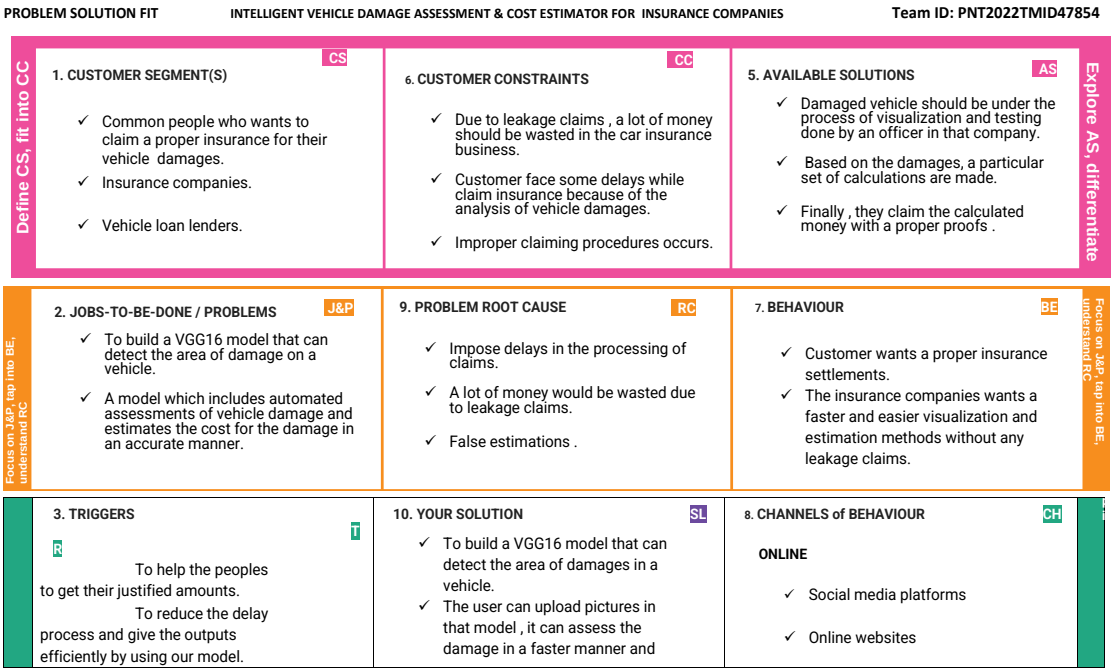
3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a CNN model that achieve the following constraints : • The developed model analyse the user uploaded picture correctly and detect the area of damage. • After that it automatically estimate the cost of damage . • The above steps are done in a faster and accurate manner .
2.	Idea / Solution description	To design a VGG16 model that will do the following actions : • To build a VGG16 model that can detect the area of damage of an vehicle. • The user will give the input in terms of picture of the damaged area .So that it can detects and assess the damaged area. • From that analysis , our model can estimate the cost of damage.
3.	Novelty / Uniqueness	• Our model will have the capability to analyse a small scratches also. • Faster mechanism • The outputs are cent percent accurately. • By our model, the insurance companies will get a lot of advantages in the area of security visualization and assessments.
4.	Social Impact / Customer Satisfaction	• Customer will get a proper insurance claims. • Fraudulent activities are avoided. It helps the customer to increase their confident level. • Delays are reduced . • This model can also be used by lenders if they are underwriting a car loan, especially for a used car.
5.	Business Model (Revenue Model)	• This model can attract a lot of insurance based companies and also car loan lenders for made a proper and faster assessment process.
6.	Scalability of the Solution	• By using this system, the leakage claims and underwriting claim problems will be avoided. • Faster processing of claims can reduce the manual delays.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Through Website Through Gmail
FR-2	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-3	**User details**	Users are required to register their personal details. like name, age, date of birth, driving license, car number etc.
FR-4	**User requirements**	They used to have the damaged car images in different angles. The car's information on the initial stages are also be provided.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This app is used in case of any uncomfortable scenario. Users have assistance from the demo section.
NFR-2	**Security**	It will have regular thread checks also protected to be privately maintained by extra verification that needed to protect the data. The system identifies the user who are in the domain to retrieve the data and the particular only able to see the data.
NFR-3	**Reliability**	This application must perform without failure in 90 percent of use cases during a month.it is more reliable.
NFR-4	**Performance**	The performance on the hand improved upto 1000 user on a very reliable basis.
NFR-5	**Availability**	As all are cloud oriented services the user can access the application 24/7.
NFR-6	**Scalability**	It is scalable to increase the number of user to be beneficial. It is scalable because of producing good results.

4. PROJECT DESIGN

5.0 Project Structure

Create a Project folder that contains files as shown below

• Dataset folder contains the training and testing images for training our model.

• We are building a Flask Application that needs HTML pages stored in the templates folder and a python script app.py forserverside scripting

• weneed themodelwhichissavedandthesavedmodel inthis contentis anutrition.h5

• templatesfoldercontainshome.html,image.html, imageprediction.htmlpages.

• Statisfolderhadthecssandjsfileswhicharenecessaryforstyling thehtmlpageandforexecutingtheactions.
•Uploadsfolderwillhavetheuploadedimages(whicharealready tested).

• Sample_imageswillhavetheimageswhichareused totestor upload.

• Trainingfoldercontainsthe trainedmodelfile.

5.1 Project flow

- The user interacts with the UI (User Interface) and give the image as input.
- Then the input image is then pass to our flask application,
- And finally with the help of the model which we build we will classify the resultand showcase it on the UI.
- To accomplish this, we have to complete all the activities and tasks listed below

Data Collection.

Collect the dataset or Create the dataset

Data Preprocessing.

- Import the ImageDataGenerator library
- Configure ImageDataGenerator class

ApplyImageDataGenerator functionality to Trainset and

Testset

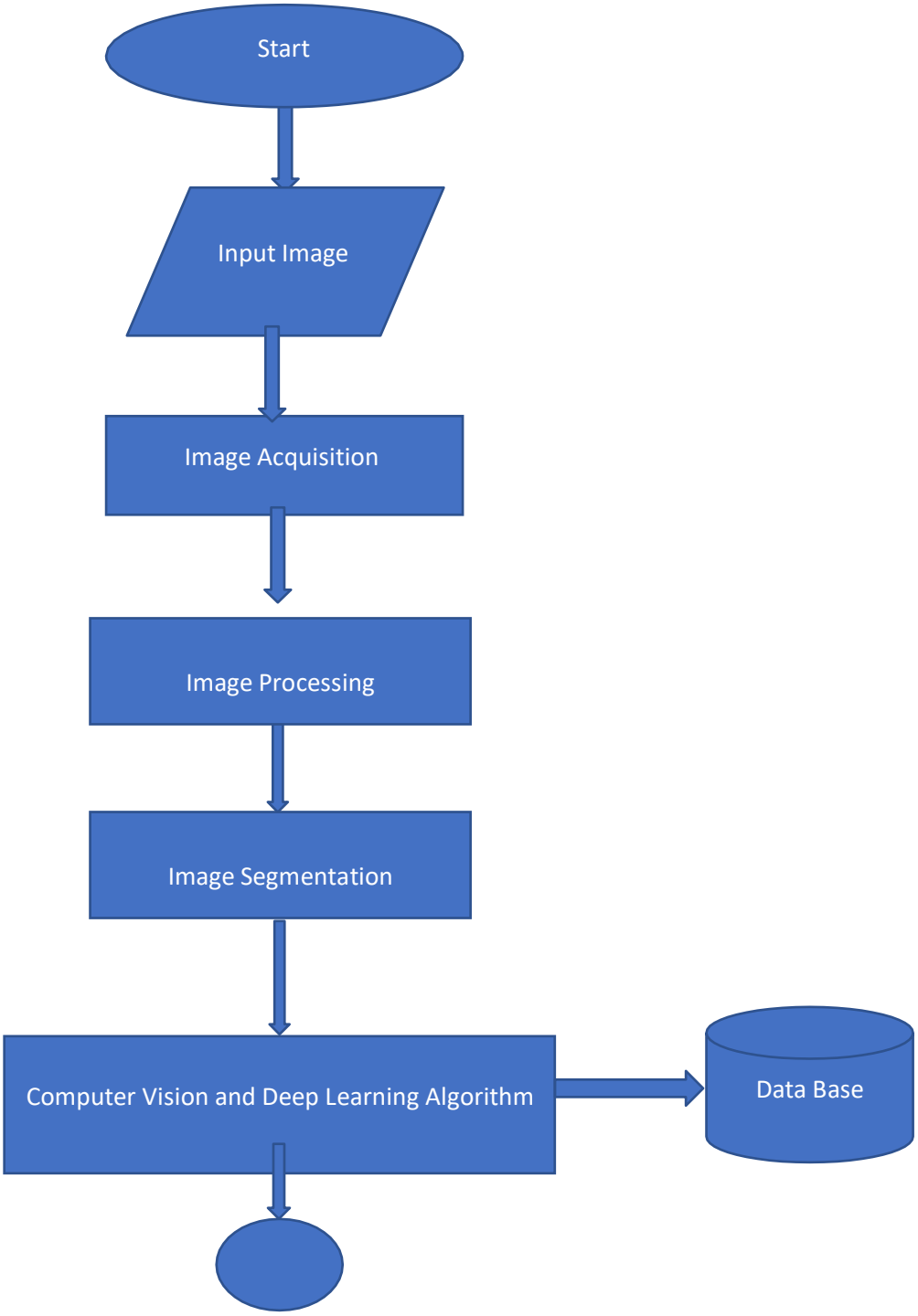
Model Building

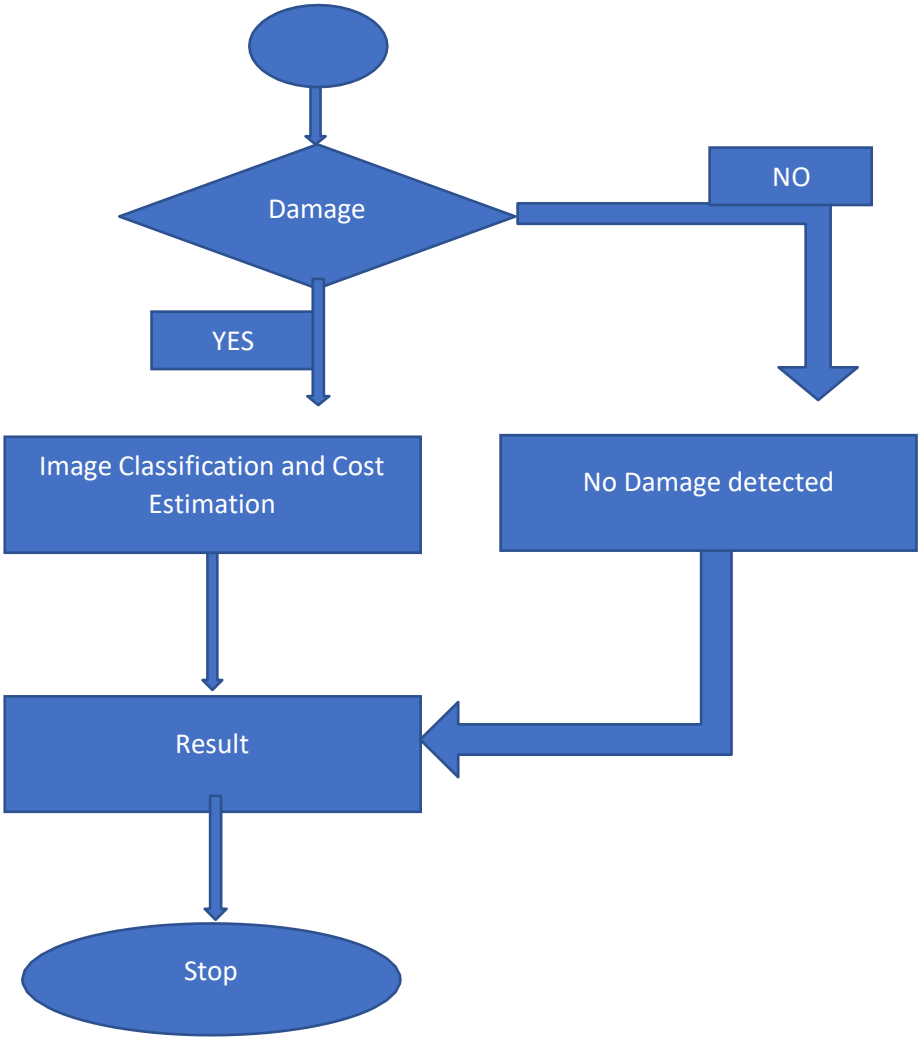
- Import the model building Libraries
- Initializing the model
- Adding Input Layer
- Adding Hidden Layer
- Adding Output Layer
- Configure the Learning Process
- Training and testing the model
- Save the Model

Application Building

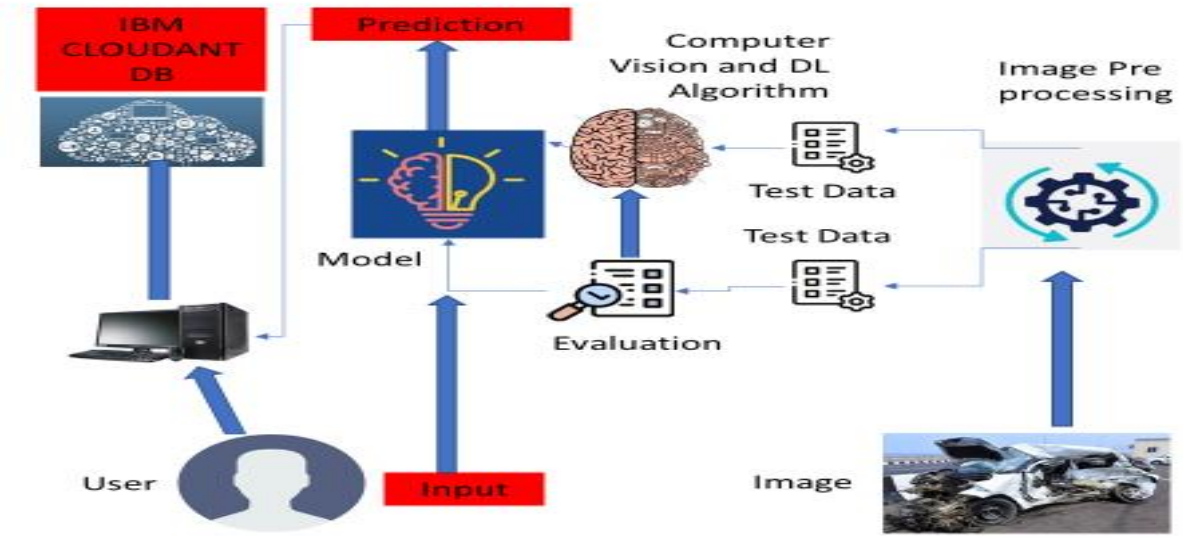
- Create an HTML file
- Build Python Code

5.2 Data Flow Diagram

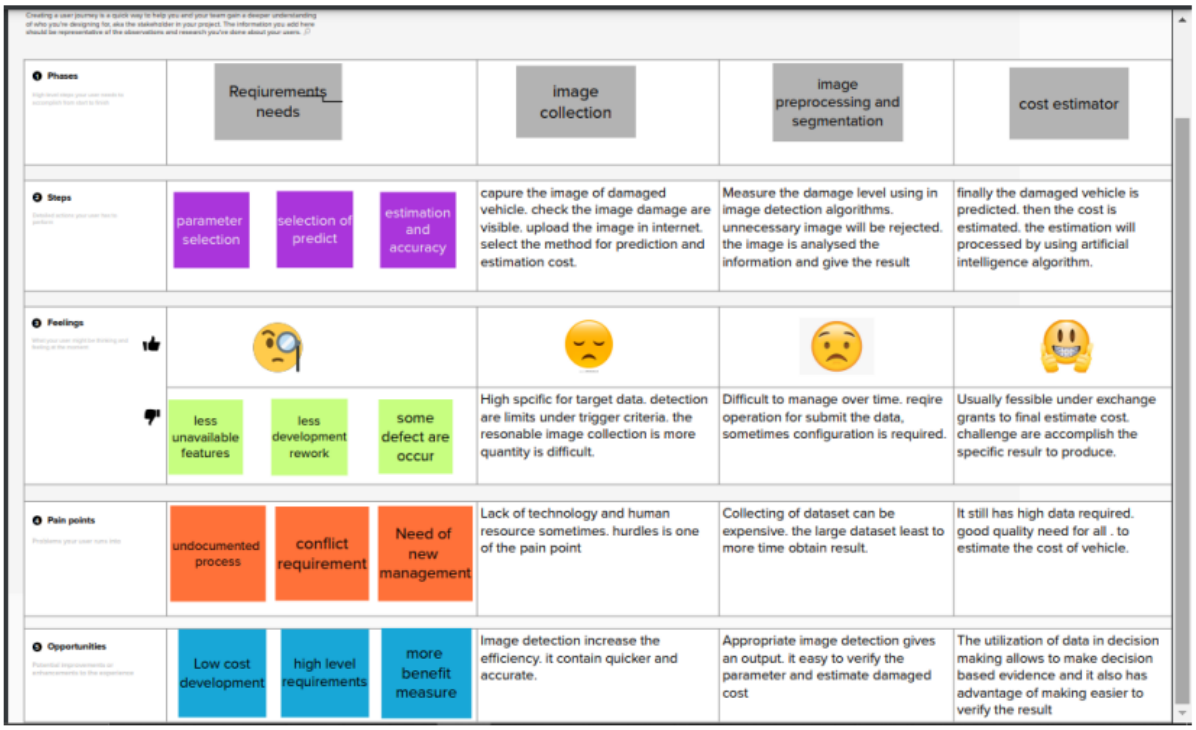




5.3 Architecture



5.4 Customer Journey



Phases	Reqiurements needs	image collection	image preprocessing and segmentation	cost estimator
Steps	parameter selection; selection of predict; estimation and accuracy	capure the image of damaged vehicle. check the image damage are visible. upload the image in internet. select the method for prediction and estimation cost.	Measure the damage level using in image detection algorithms. unnecessary image will be rejected. the image is analysed the information and give the result	finally the damaged vehicle is predicted. then the cost is estimated. the estimation will processed by using artificial intelligence algorithm.
Feelings	🧐	😔	😟	🤐
	less unavailable features; less development rework; some defect are occur	High spcific for target data. detection are limits under trigger criteria. the resonable image collection is more quantity is difficult.	Difficult to manage over time. reqire operation for submit the data, sometimes configuration is required.	Usually fessible under exchange grants to final estimate cost. challenge are accomplish the specific resulr to produce.
Pain points	undocumented process; conflict requirement; Need of new management	Lack of technology and human resource sometimes. hurdles is one of the pain point	Collecting of dataset can be expensive. the large dataset least to more time obtain result.	It still has high data required. good quality need for all . to estimate the cost of vehicle.
Opportunities	Low cost development; high level requirements; more benefit measure	Image detection increase the efficiency. it contain quicker and accurate.	Appropriate image detection gives an output. it easy to verify the parameter and estimate damaged cost	The utilization of data in decision making allows to make decision based evidence and it also has advantage of making easier to verify the result

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	2	High	team member 2, team member 4
Sprint-1	Login	USN-2	As a user, I will receive a confirmation email once I have registered for the application	1	High	team member 3 and team member 4
Sprint-1	Dashboard	USN-3	As a user, I can register for the application through Facebook	1	High	team member 1 and team member 3
Sprint-2	Details about insurance company	USN-4	As a user, I can register for the application through Gmail	1	low	team member 3
Sprint-1	repeated logins and logout	USN-5	As a user, I can log into the application by entering email & password	2	medium	team member 1 team

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	2	High	team member 2, team member 4
Sprint-1	Login	USN-2	As a user, I will receive a confirmation email once I have registered for the application	1	High	team member 3 and team member 4
Sprint-1	Dashboard	USN-3	As a user, I can register for the application through Facebook	1	High	team member 1 and team member 3
Sprint-2	Details about insurance company	USN-4	As a user, I can register for the application through Gmail	1	low	team member 3
Sprint-1	repeated logins and logout	USN-5	As a user, I can log into the application by entering email & password	2	medium	team member 1 and team member 2

(Explain the features added in the project along with code)

7.0 Pre-requesties

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 so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. Forthis

project, we will be using a Jupyter notebook and Spyder.

To install the Anaconda navigator and to know how to use Jupyter Notebook & Spyder using Anaconda watch the video

1. 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.

      Flask Basics

If you are using Pycharm IDE, you can install the packages through the command prompt and follow the same syntax as above.

7.TESTING

TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

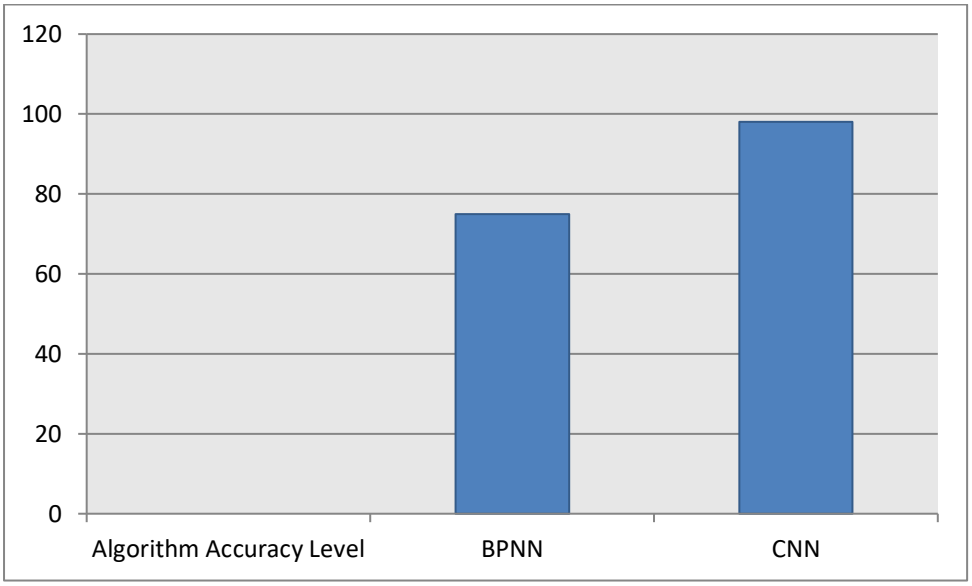
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload damaged vehicle image as a input	Detecting object and analyze for claim insurance	Details are stored in a database.

USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT

8.RESULTS

PERFORMANCE METRICS



10ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Digitalized claim process makes easy to use
- Give the accurate result of the damaged vehicle
- Helps the insurance company to analyze the damaged vehicle and also payment process.

DISADVANTAGE

- It will take more time to claim the insurance in manual process
- Because of incorrect claims, the company behaves badly and doesn’t make payments currently.
- Poor customer support

11.CONCLUSION

In this research proposal, a neural network-based solution for automobile detection will be used to address the issues of automotive damage analysis and position and severity prediction. This project does several tasks in one bundle. The method will unquestionably assist the insurance firms in conducting far more thorough and systematic analyses of the vehicle damage. Simply sending the system a photograph of the vehicle, it will evaluate it and determine whether there is damage of any type, where it is located, and how severe it is.

12.FUTURE SCOPE

In future work, need to use several regularisation methods with a big dataset in our next work. Anticipate the cost of a car damaged component more accurately and reliably if we have higher quality datasets that include the attributes of a car (make, model, and year of production), location data, kind of damaged part, and repair cost. This study makes it possible to work together on picture recognition projects in the future, with a focus on the auto insurance industry. The study was able to accurately validate the presence of damage, its location, and its degree while eliminating human bias. These can be further enhanced by adding the on the fly data augmentation approaches.

13.APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request,session
from cloudant.client import  Cloudant
import cv2
client = Cloudant.iam("eb55a2b7-ae45-4df8-8d1c-69c5229ffdbe-
bluemix","YzG5FZg9Vs_HScOBZaWyVXm7PpNjbPrmPaPMfHx7w3X9",connect=
True)
my_database = client.create_database("database-dharan")
app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/userhome")
def userhome():
    return render_template('userhome.html')
@app.route("/addamount")
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')


@app.route("/user")
def user():


    return render_template('user.html')
@app.route("/newuse",methods=['GET','POST'])
def newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {
```

```
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
            url = my_database.create_document(data)
            return render_template('goback.html', data="Register, please login using your
details")
        else:
            return render_template('goback.html', data="You are already a member, please
login using your details")
@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
        if request.method == 'POST':
            user = request.form['_id']
            passw = request.form['psw']
            print(user, passw)
            query = {'_id': {'$eq': user}}
            docs = my_database.get_query_result(query)
            print(docs)
            print(len(docs.all()))
            if (len(docs.all()) == 0):
                return render_template('goback.html', pred="The username is not found.")
            else:
                if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):
                    return render_template("userhome.html")
                else:
                    return render_template('goback.html',data="user name and password
incorrect")
```

```
@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        file.save('static/Out/Test.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        result = classifierLoad.predict(test_image)
        result1 = ''
        if result[0][0] == 1:
            result1 = "front"
        elif result[0][1] == 1:
            result1 = "rear"
        elif result[0][2] == 1:
            result1 = "side"
        file = request.files['fileupload1']
        file.save('static/Out/Test1.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('level.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test1.jpg', target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test1.jpg')
        # test_image = image.img_to_array(test_image)
```

```
            test_image = np.expand_dims(test_image, axis=0)
            result = classifierLoad.predict(test_image)
            result2 = ''
            if result[0][0] == 1:
                result2 = "minor"
            elif result[0][1] == 1:
                result2 = "moderate"
            elif result[0][2] == 1:
                result2 = "severe"
            if (result1 == "front" and result2 == "minor"):
                value = "3000 - 5000 INR"
            elif (result1 == "front" and result2 == "moderate"):
                value = "6000 8000 INR"
            elif (result1 == "front" and result2 == "severe"):
                value = "9000 11000 INR"
            elif (result1 == "rear" and result2 == "minor"):
                value = "4000 - 6000 INR"
            elif (result1 == "rear" and result2 == "moderate"):
                value = "7000 9000 INR"
            elif (result1 == "rear" and result2 == "severe"):
                value = "11000 - 13000 INR"
            elif (result1 == "side" and result2 == "minor"):
                value = "6000 - 8000 INR"
            elif (result1 == "side" and result2 == "moderate"):
                value = "9000 - 11000 INR"
            elif (result1 == "side" and result2 == "severe"):
                value = "12000 - 15000 INR"
            else:
                value = "16000 - 50000 INR"
            return render_template('userhome.html', prediction=value)
        if __name__ == '__main__':
app.run(debug=True, use_reloader=True)
```

GitHub

https://github.com/IBM-EPBL/IBM-Project-45850-1660732782

Project Demo Link

https://drive.google.com/file/d/1ZUVguKiF1PauvXa0XcO98UPd2y5lFn9h/view?usp=sharing