Project On

Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies

powered By IBM India

Submitted By

PRINCE ROHAN J

SURYA D

SWAMINATHAN J

SURESH S

Project ID: PNT2022TMID00513

of

Department of INFORMATION TECHNOLOGY

ST.JOSEPH'S COLLEGE OF ENGINEERING

OMR, Chennai - 600119

College Mentor: **DIVYA J**

Industrial Mentor: **SWATHI**

INDEX

1. INTRODUCTION

1.1. Project Overview

The project "Intelligent Vechile Damage Assessment and Cost Estimator for Insurance Companies" is a responsive web application powered by aritifical Intelligence and IBM Watson Cloud. Deep Learning model is trained with the various damaged car images in various views and the VGG16 from the TensorFlow library is used for the better Deep Learning model architecture. An attractive front end can be developed using HTML and CSS. The pages such as Index.html , login.html, logout.html, register.html and prediction.html are created and embedded with the IBM cloud databse using python framework called flask. The web application takes the image input and estimate the cost for the insurance companies based on the damages in the car.

1.2. Purpose

The project is based on the domain of Artificial Intelligence and powered by the IBM watson cloud. A responsive web application can be developed using the HTML and CSS which is connected to waston cloud. In the cloud, a database service by availing the service Instance of the IBM cloud and the database API key is collected and connected with the front-end using flash which is an python framework for desiging the backend. Pages such as index.html, login.html, logout.html and prediction.html are used to interact with the web application. The user can register and the data of the user is saved in the databse of the IBM cloud, during the time of login, the login ID is compared with the ID in the databse and allow the user to the next page. The Deep Learning model is build using the VGG16 which is present in the keras library and the model is trained with the images of mulitple car with various level cum types of damages. The model is deployed in the back-end using the flask and the prediction.html page is setted to collect the image from the user. The prediction algorithm is used treat the image and estimated the cost for the user. The project is based on the various components which helps to handle the back - end and Front - end. Then front - end is build using html and css which is connectedback - end which is build using the python and IBM cloud. The project is powered by the IBM Watson cloud and is based in the artificial intelligence field. With the use of HTML and CSS and the Waston Cloud, a responsive web application may be created. The database API key is gathered and connected with the front-end using flash, which is a python framework for designing the backend.

2. LITERATURE SURVEY

2.1 Existing problem

1. Damage Assessment of a vehicle and Insurance Reclaim: This paper presents a system using CNN and image classification to assess the severity of damage to an automobile, which takes a user's input as an image to test the severity of the damage, which happens in two steps. The first step is image classification, where the user's input is used by the neural network to determine whether or not an automobile is damaged. the region and severity of the damage are determined in the second step using object detection on the flattened input that was received as the output in step one. The area may be the back, the front, or the side, and the severity may be classified as minor, moderate, or major. A report is filed and delivered to the user and the insurance company when the R-CNN network determines the extent of the damage. With little human contact, the user will be able to receive payment based on the results of the models.

2. Convolutional Neural Networks for vehicle damage detection: In this paper, a model for detecting vehicle damage is created, and it is divided into twelve categories. A deep learning model that can accurately detect and classify vehicle damages is created and evaluated in a specially designed light street, indicating that strong reflections complicate the detection performance. The proposed model outperforms other existing models in the classes Bend and Cover Damage. FSSD with Darknet-53 and YOLO v3 with Darknet-53 yield the best results. The drawback of the proposed approach is the robustness against different light conditions

3. Car Damage Assessment for Insurance Companies: In this paper a neural network-based solution for car detection, managing the problem of car damage analysis, prediction of car damage location and severity of the damage is proposed. The proposed system is intended to help insurance companies to analyze car damage a lot more successfully and well organized, and it quickly performs car damage detection by sending the image containing a damaged car for visual inspection. This system utilizes a machine learning approach along with computer vision to decide the damage analysis, the location of the damage as well as

the severity of the damage.

2.2 PROBLEM STATEMENT DEFINITION

4. Assessing Car Damage with Convolutional Neural Networks: This study focuses on automotive damage estimation, with auto insurers as their main potential clients. Three different Transfer Learning techniques are employed to do this, each of which identifies the existence, location, and degree of damage. Convolutional Neural Networks, which are adapted to maximize accuracy, serve as the foundation for the algorithms used. Each approach is analyzed and varying degrees of accuracy were achieved across different models deployed ranging from 68% to 87%. In this work, accuracy as high as 87.9% was attained. This study improves a number of existing methods and creates opportunities for collaboration in image recognition, notably in the field of auto insurance.

5. Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning: This paper proposes deep learning-based methods for the classification of car damage types - MobileNet to classify vehicle damage into three groups: medium damage, enormous damage, and no damage. The extent of the damage to the vehicle determines its severity, ranging from medium to huge. The damage categories are based on typical damage kinds including shattered glass, dents on the front or back, damaged lamps or bumpers, etc. Automation in real-time applications, however, faces several challenges. Instead of capturing a picture of a car in real time, users can upload fake pictures. Making fake photos can involve using image-editing software to cover up flaws, getting images from the internet, or even taking screenshots of other devices' screens. To deal with these kinds of fake photographs, a hybrid strategy is also suggested in this research. To determine whether an image has been altered or is a screenshot, metadata analysis, and image editing software signature detection are used. It is suggested that moiré effect detection be used to determine whether an image was captured from the screen of another device, such as a computer screen when a mobile phone was used to snap a photo of an automobile.

6. Deep Learning Based Car Damage Classification and Detection: In this paper, they address the problem of vehicle damage classification/detection, which can be used by insurance companies to automate the process of vehicle insurance claims. With the adoption of fast, scalable, and end-to-end trainable convolutional neural networks, it is now technically feasible to recognize vehicle damages using deep convolutional networks. Various online sources containing different types of vehicle damage were manually collected and annotated. Using CNN models pre-trained on the ImageNet dataset and other techniques to improve the performance of the system, we achieved top accuracy of 96.39%, significantly better than the current results. In addition, they used a state-of-the-art YOLO object detector to detect the damaged region, achieving a maximum map score of 77.78% on the held-out test set, demonstrating the model's ability to recognize different vehicle damages. Furthermore, the paper proposes a pipeline for more robustly identifying vehicle damage by combining classification and detection tasks.

7. Car damage detection and classification: In this paper, a CNN model is developed and trained on the ImageNet dataset. After fine-tuning the dataset, transfer learning with L2 regularization is applied. In the proposed system, a Pre-trained VGG model not only detects the damaged part of a car but also assesses its location and severity. With the use of transfer learning and L2 regularisation, the proposed system achieves an accuracy of 95.22% of VGG19 and 94.56% of VGG16 in damaged detection, 76.48% of VGG19 and 74.39% of VGG16 in damage localization, and 58.48% of VGG19 and 54.8% of VGG16 in damage severity.

2.3.REFERENCES

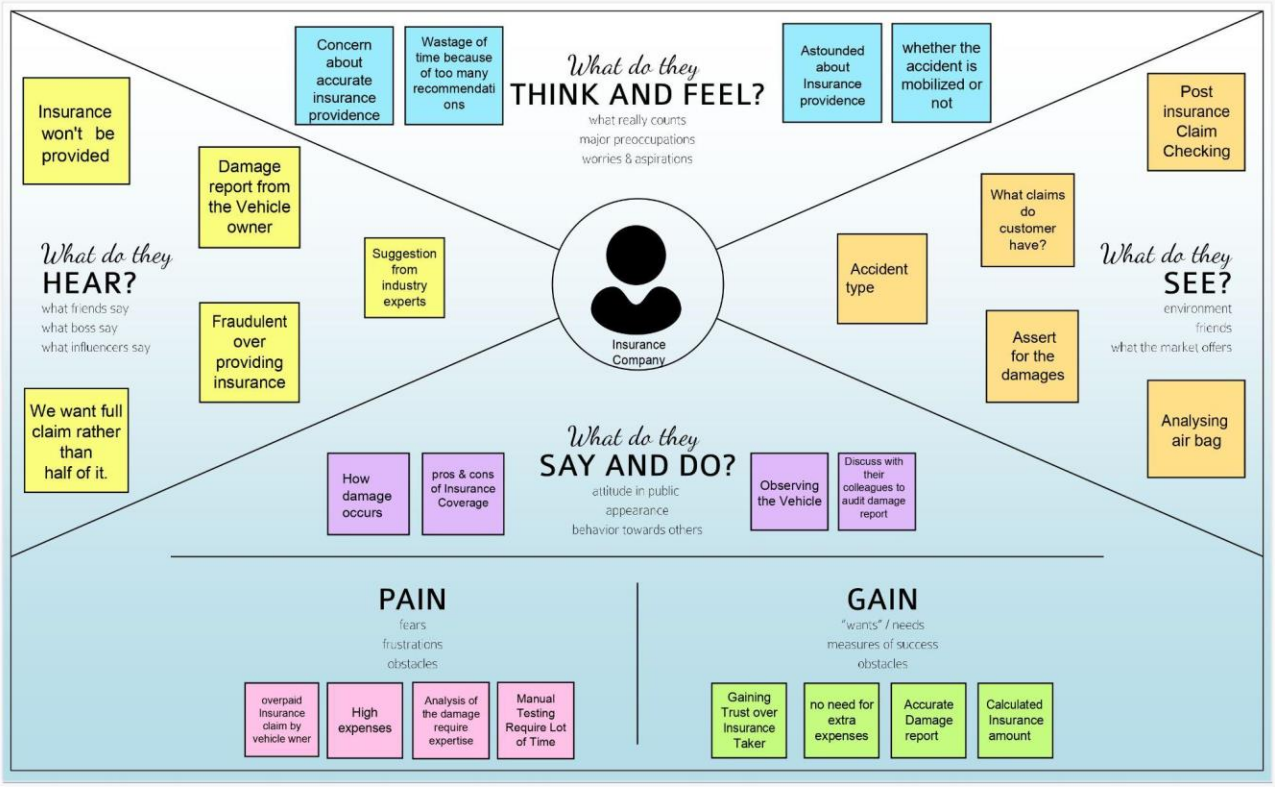
8. Vaibhav Agarwal, Utsav Khandelwal, Shivam Kumar, Raja Kumar, Shilpa M, "Damage Assessment Of A Vehicle And Insurance Reclaim", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 4, pp.e197-e201, April 2022, Available at :http://www.ijcrt.org/papers/IJCRT2204483.pdf 2. R. E. van Ruitenbeek and S. Bhulai, “Convolutional Neural Networks for vehicle damage detection,” Machine

Learning with Applications, vol. 9. Elsevier BV, p. 100332, Sep. 2022. doi: 10.1016/j.mlwa.2022.100332. 3. Mandara G S and Prashant Ankalkoti, “Car Damage Assessment for Insurance Companies,” International Journal of Advanced Research in Science, Communication and Technology. Naksh Solutions, pp. 431–436, Jun. 23, 2022. doi: 10.48175/ijarsct-5048. 4. H. Bandi, S. Joshi, S. Bhagat, and A. Deshpande, “Assessing Car Damage with Convolutional Neural Networks,” 2021 International Conference on Communication information and Computing Technology (ICCICT). IEEE, Jun. 25, 2021. doi: 10.1109/iccict50803.2021.9510069. 5. 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 (CCECE), 2020, pp. 1-5, doi: 10.1109/CCECE47787.2020.9255806. 6. M. Dwivedi et al., “Deep Learning-Based Car Damage Classification and Detection,” Advances in Intelligent Systems and Computing. Springer Singapore, pp. 207–221, Aug. 14, 2020. doi: 10.1007/978-981-15-3514-7_18. 7. P. M. Kyu and K. Woraratpanya, “Car Damage Detection and Classification,” Proceedings of the 11th International Conference on Advances in Information Technology. ACM, Jul. 2020. doi: 10.1145/3406601.3406651.

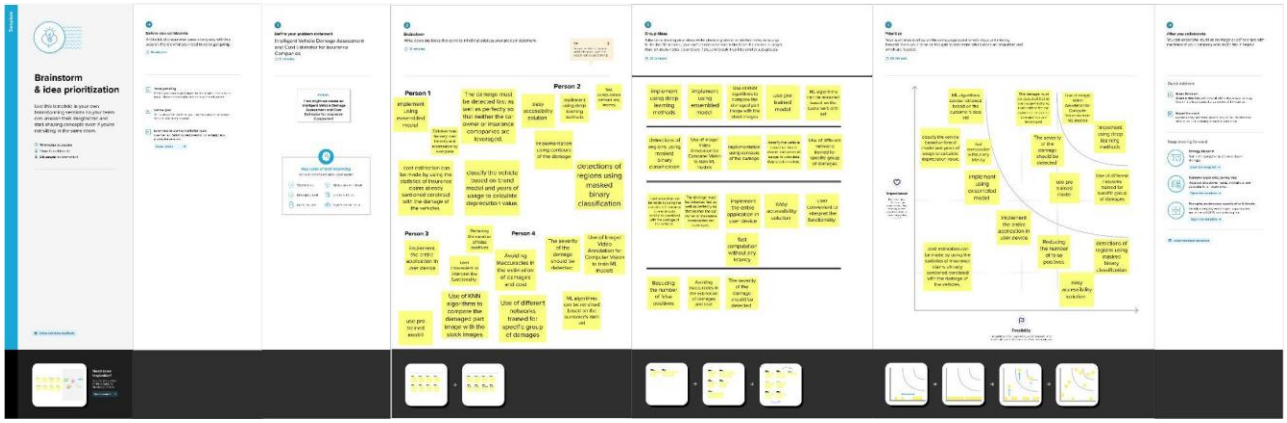
3. IDEATION & PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	To accomplish this, to create Train and Test Folders and then image preprocessing in which Import the imagedatagenerator library and applyimagedatagenerator functionality to Trainset and Testset. The third step is Model Building in which Import the model building Libraries,Adding Flatten layers then Adding Output Layer then Creating Model Object then Configure the Learning Process then Train,Save,Test The Model. Step four is Cloudant DB in which Register & Login to IBM Cloud then Create Service Instance and Credentials then Launch Cloudant DB then Create Database. The last step is Application Building in which Building HTML Pages then Build Python Code finally Run The Application
3.	Novelty / Uniqueness	▪ AI based car detection. ▪ Image processing
4.	Social Impact / Customer Satisfaction	Customer (insurance company) no need to give full amount to the policy holder. They can provide an amount based on the severity of the damage.
5.	Business Model (Revenue Model)	Subscription and advertising model
6.	Scalability of the Solution	It allows the client to avoid giving the total amount of insurance to the policyholder for a small amount of damage.

3.1 Empathy Map



3.2 Ideation and Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	To accomplish this, to create Train and Test Folders and then image preprocessing in which Import the imagedatagenerator library and applyimagedatagenerator functionality to Trainset and Testset. The third step is Model Building in which Import the model building Libraries,Adding Flatten layers then Adding Output Layer then Creating Model Object then Configure the Learning Process then Train,Save,Test The Model. Step four is Cloudant DB in which Register & Login to IBM Cloud then Create Service Instance and Credentials then Launch Cloudant DB then Create Database. The last step is Application Building in which Building HTML Pages then Build Python Code finally Run The Application
3.	Novelty / Uniqueness	▪ AI based car detection. ▪ Image processing
4.	Social Impact / Customer Satisfaction	Customer (insurance company) no need to give full amount to the policy holder. They can provide an amount based on the severity of the damage.
5.	Business Model (Revenue Model)	Subscription and advertising model
6.	Scalability of the Solution	It allows the client to avoid giving the total amount of insurance to the policyholder for a small amount of damage.

3.4 Proposed Solution Fit

Project Title: Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies | **Project Design Phase-I - Problem Solution Fit** | **Team ID: PNT2022TMID00513**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Drivers **aged between 25 and 65** are the most common age group of customers for car insurance.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Customers may feel that our website is **not trustworthy** due to some other scam websites.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Just by sending the image of damaged car to our website, customer gets the details of amount to be **claimed in a minute** rather than days if it is inspected visually. There won't be any claim leakage problems.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Customers limit themselves from claiming insurance for minor damages because of **claims leakage** (Difference between the final settled amount paid out by an insurer and the amount that they could've paid had the claims process been more efficient).

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The real problem arises when the customer has severe damage on the car and they get **minimum amount** than expected. Since many people are involved at various stages of a claim, there is lack of visibility which makes the process to **slow down** and **over-complicated** at different stages.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. Directly related: find the right solar panel installer, calculate usage and benefits; Indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Whenever the customer has damage on the car, they **meet the insurer** and apply for claim amount. As this process is time consuming, the customers **search** for car insurance websites to claim the amount. They upload the image of damaged car and get the details of claim amount within fraction of seconds.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Being **transparent** to the customers by not making any false guarantees

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

We should prove that our website is better than others by providing good customer support, gaining the customer trust and provide customer satisfaction.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

The aim of this project is to estimate the cost of damaged car accurately by detecting the area of damage, categorizing the damage with precision in a **fast and intelligent manner**. It can be used by insurance companies for faster processing of claims if users can upload pictures.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

- Select the model of damaged car.
- Select the city where you live.
- Upload the image of damaged

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Meeting the insurer.
- Filling application forms.
- Submitting the required documents.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registration	Download the app Registration Done via Gmail Create an account Follow the instructions
FR-2	User Confirmation	Confirmation through Email Confirmation through OTP
FR-3	Interface	Understandable Interface for user to operate
FR-4	Accessing datasets	Details about user Details about vehicle Details about insurance companies
FR-5	Mobile application	AI and camera sensor in the field can be access by mobile application.

4.2 NON FUNCTIONAL REQUIREMENTS

Non-functional Requirements:

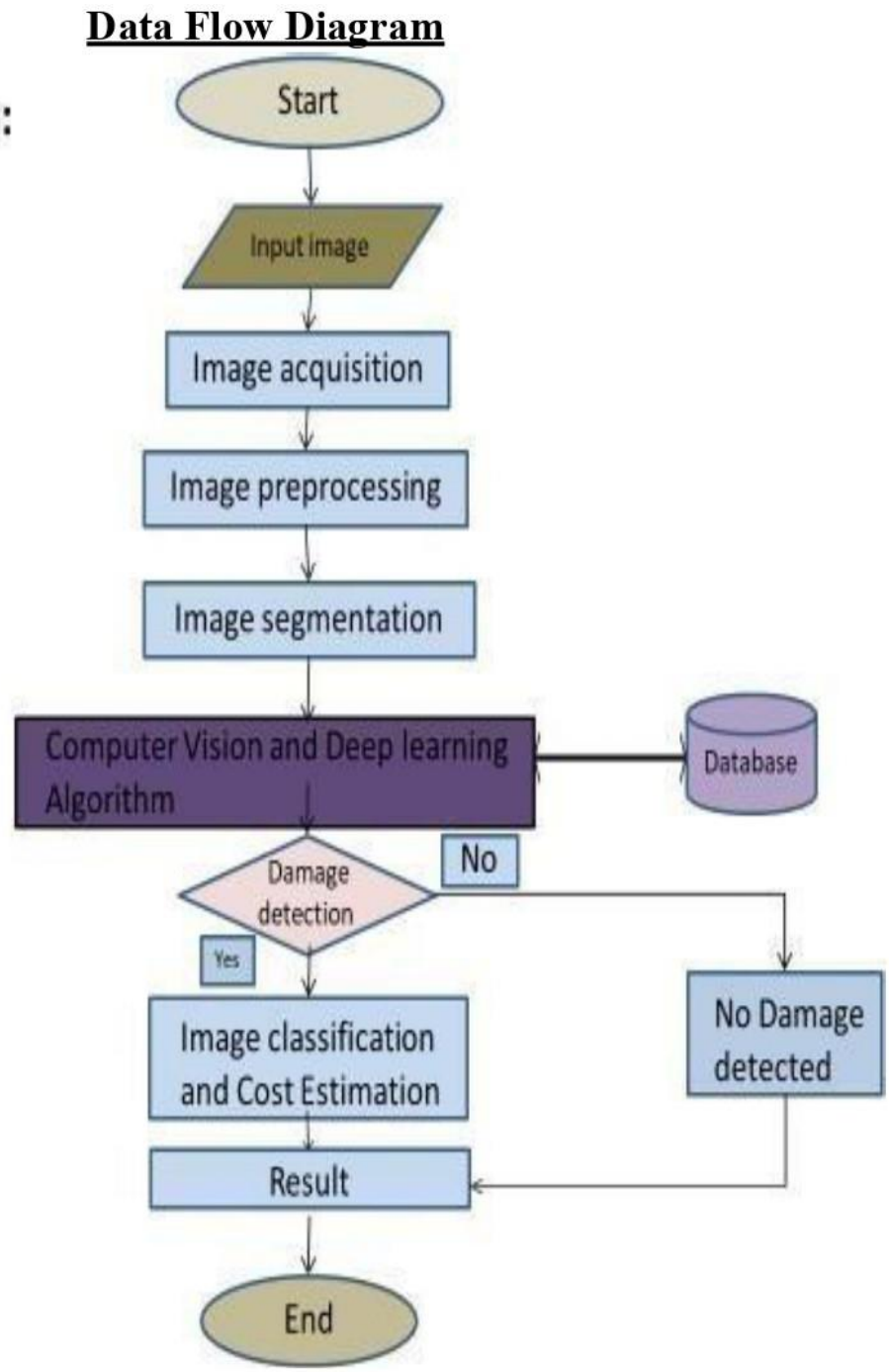
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The smart claiming system for vehicle damage insurance in bank companies
NFR-2	**Security**	We have designed this project to user easy to claim the insurance .
NFR-3	**Reliability**	This project will help the user to claim the insurance cost based on vehicle damage. It gives the exact value to user. This helps user to get correct cost without any failure.
NFR-4	**Performance**	AI devices and sensors are used to indicate the user to estimated the cost of the vehicle.AI camera to scan the damaged vehicle and gives exact cost insurance to user.
NFR-5	**Availability**	This application is designed for all devices and also available in apk.
NFR-6	**Scalability**	This project is more scalability in our present and future uses to estimate the cost exactly to user.

5. PROJECT DESIGN

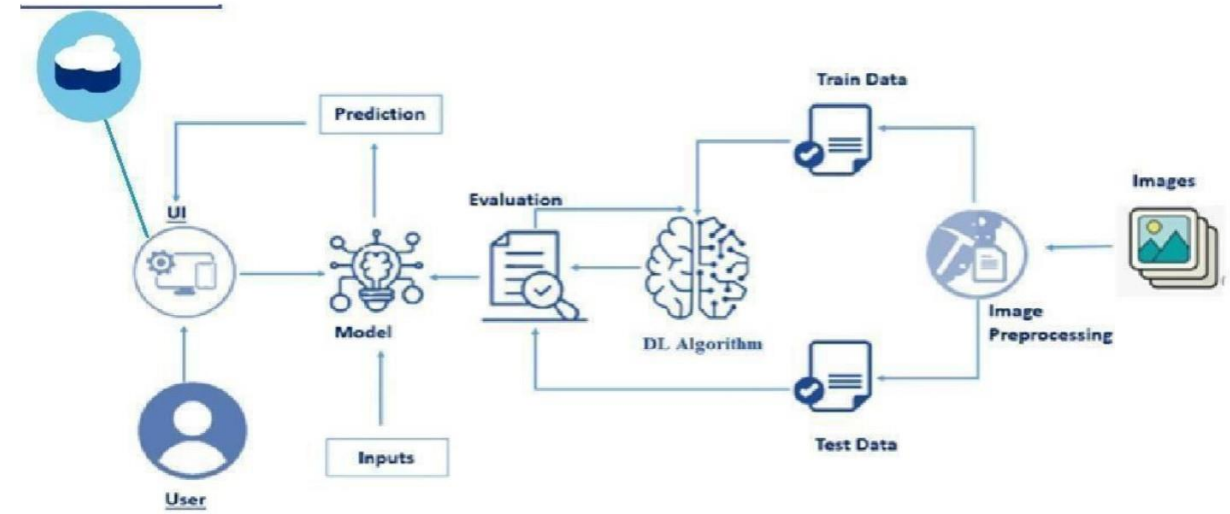
5.1 DATA FLOW DIAGRAM

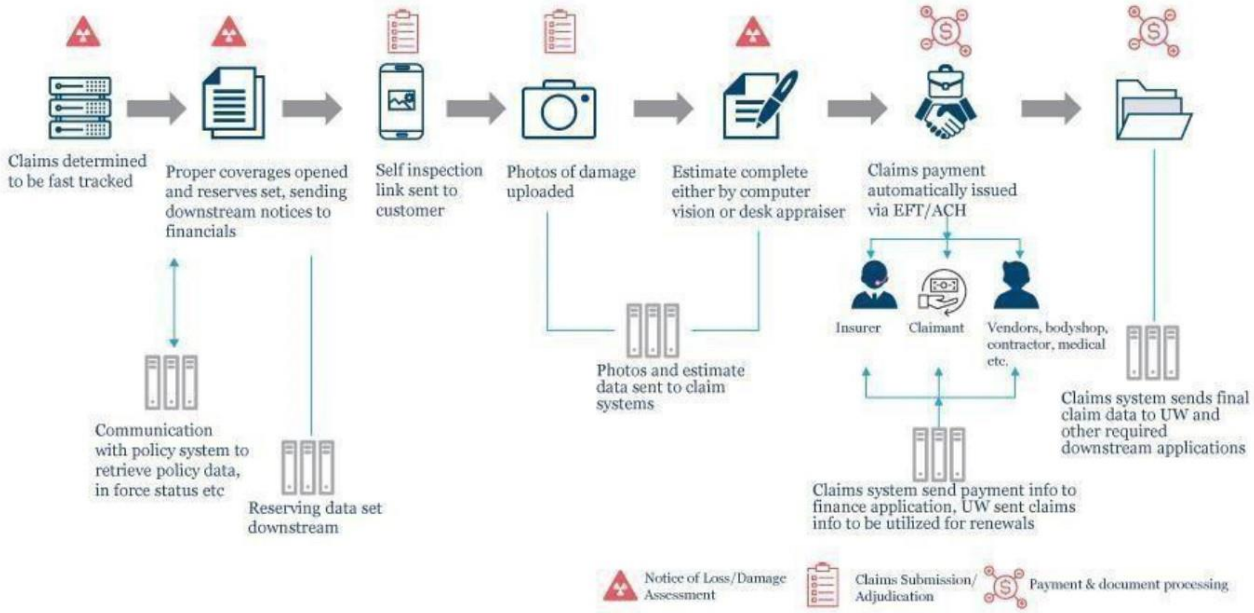
Data Flow Diagram:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technology Architecture:





5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard by entering valid credentials	High	Sprint-1
Customer Details	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer Uses	Dashboard	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-4
Customer Options	Details about insurance companies	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Facebook Gmail	Medium	Sprint-1
Customer usage	Login	USN-5	As a user, I can log into the application by entering email & password	I can log in and view my dashboard at my demand on any time	High	Sprint-1
Customer needs to do	Dashboard	USN-6	As a user I must capture images of my vehicle and upload it into the web portal	I can capture the entire vehicle and upload	High	Sprint-2
Customer (Web user)	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the cost estimated	I can get the estimated insurance cost	High	Sprint-3
Customer Care Executive	Details about Estimated cost Based on damage	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	I can have smooth user experiences and all the issues raised is sorted	Medium	Sprint-4
Administrator	Details about Estimated cost Based on damage	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	I can finish the work without any problems	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and 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 details of name, email, cars etc. verifying my Gmail account and creating new account with password	7	HIGH	TM-1,4
Sprint-1	Login	USN -2	As a user, entering my email, and password, and confirming my password, I can login to myaccount.	7	HIGH	TM-1,4
Sprint-1	Dashboard	USN-3	As a user, I can clearly see data, point, graphs, charts and trends of my previous activity and global activity related to my views	2	LOW	TM-1,4
Sprint-2	Details about insurance company	USN-4	As a user, I can register for the application through Gmailand account id.	8	MEDIUM	`TM-2,3
Sprint-1	repeated logins and logout	USN-5	As a user, I can log in and view my dashboard at my demand on any time	4	HIGH	TM-1,4
Sprint-2	Webpage	USN-6	As a user, I must enter all details of car, accident, capture images of my vehicle and upload it into the web portal.	12	HIGH	TM-2,3
Sprint 3	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the Cost estimated.	20	HIGH	TM-1,2
Sprint 4	Provide friendly and efficient	USN-8	As a user, I need to get support from developers in case of	10	MEDIUM	TM-1,2,3

	customer support and sort out the queries.		queries and failure of service Provided by chat-box, mail orcall.			
Sprint 4	overview the entire process and act as a bridge between user and developer	USN-9	As a team member, we need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	10	HIGH	TM-1,2,3

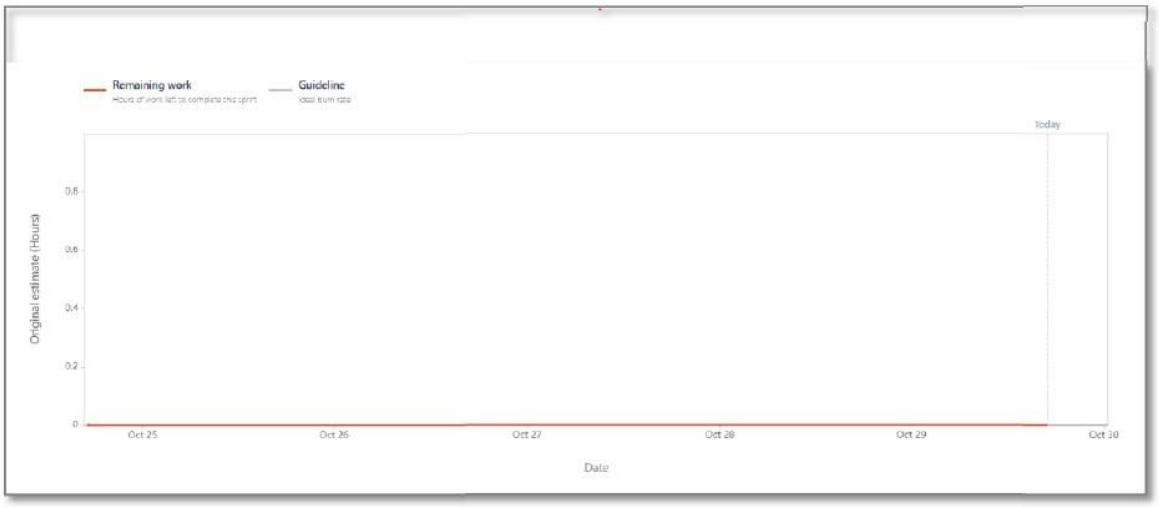
6.2. SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	-	-
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	-	-
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	-	-

Velocity:

$$AV = \frac{SPRINT\ DURATION}{VELOCITY} = \frac{20}{6} = 3.33$$

Burn-down Chart:



7. CODING & SOLUTIONING

7.1 FEATURE 1

```
client = Cloudant.iam("1c6f917d-87ac-491b-90a0-
6e3ae5b5daca-
bluemix","tYJcUyVJYs3WrxF_1absTN4RXrbdQ_RDWBRUy9BX-
28c",connect=True)
database =
 #load model
 model1 = load_model('V:\\WorkSpace\\IBM-Project-
 23426- 1659882722\\Final
 Deliverables\\model\\body.h5') model2 =
 load_model('V:\\WorkSpace\\IBM-Project-23426-
```

The feature 1 gives access to the trained deep learning models for predicting mulitple damages in various areas in the vechile and connected with the IBM Waston Database for storing the user data.

7.2 FEATURE 2

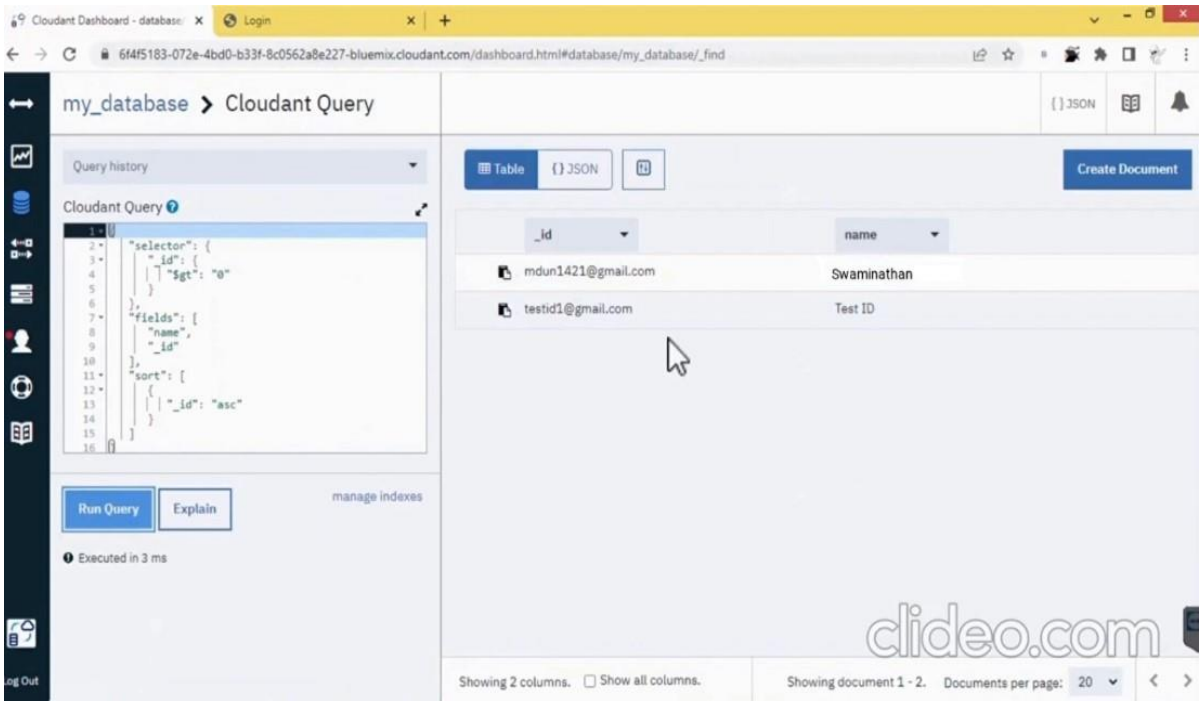
```
1
2
3
4
5
6
7
8
        img =
        load_img(filepath,target_size=(224,2

        prediction1 =
        np.argmax(model1.predict(img_data)) prediction2
        = np.argmax(model2.predict(img_data))

        index1 = ['front','near','side']
```

feature 2 enables the web application to predict the incomming image from the user into the given labels. The code gets the image, convert into pixcels and load into the model. Based on the predicted results, the algorithm will returns the value as the estimated cost.

7.3 **DATABASE SCHEMA**

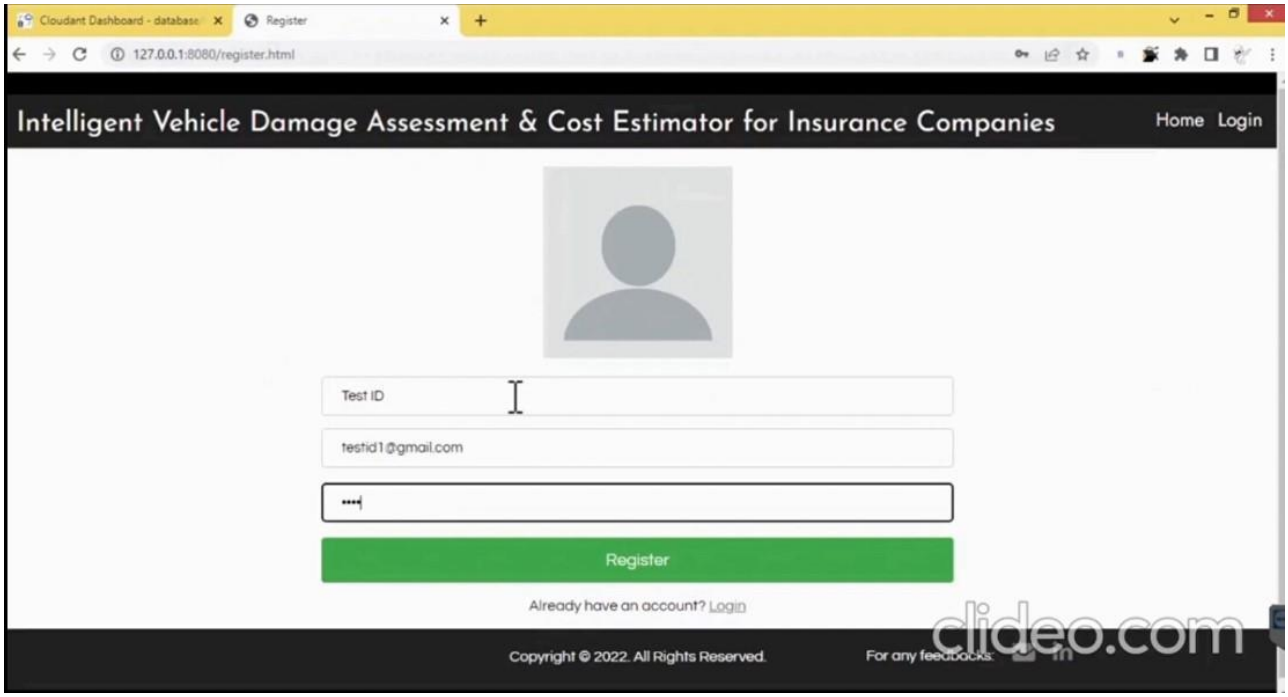


8.TESTING

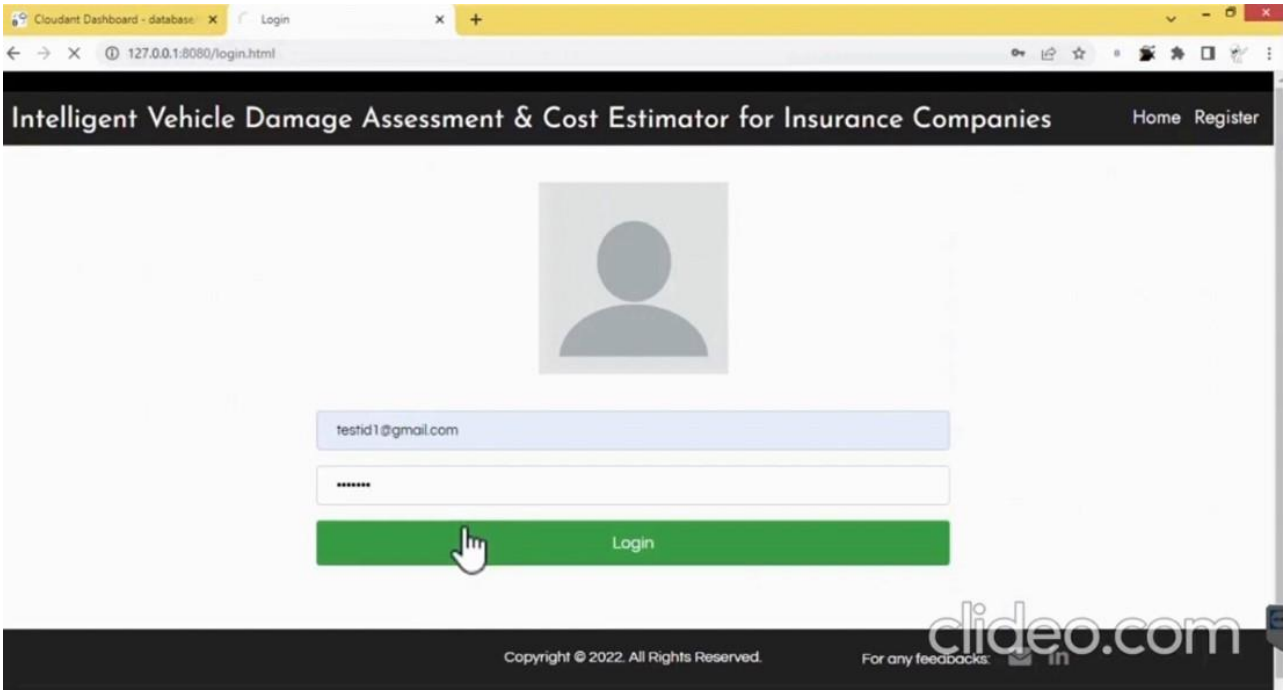
8.1. TEST CASES

1. User Login and Registration test
2. Database Update test
3. Prediction test

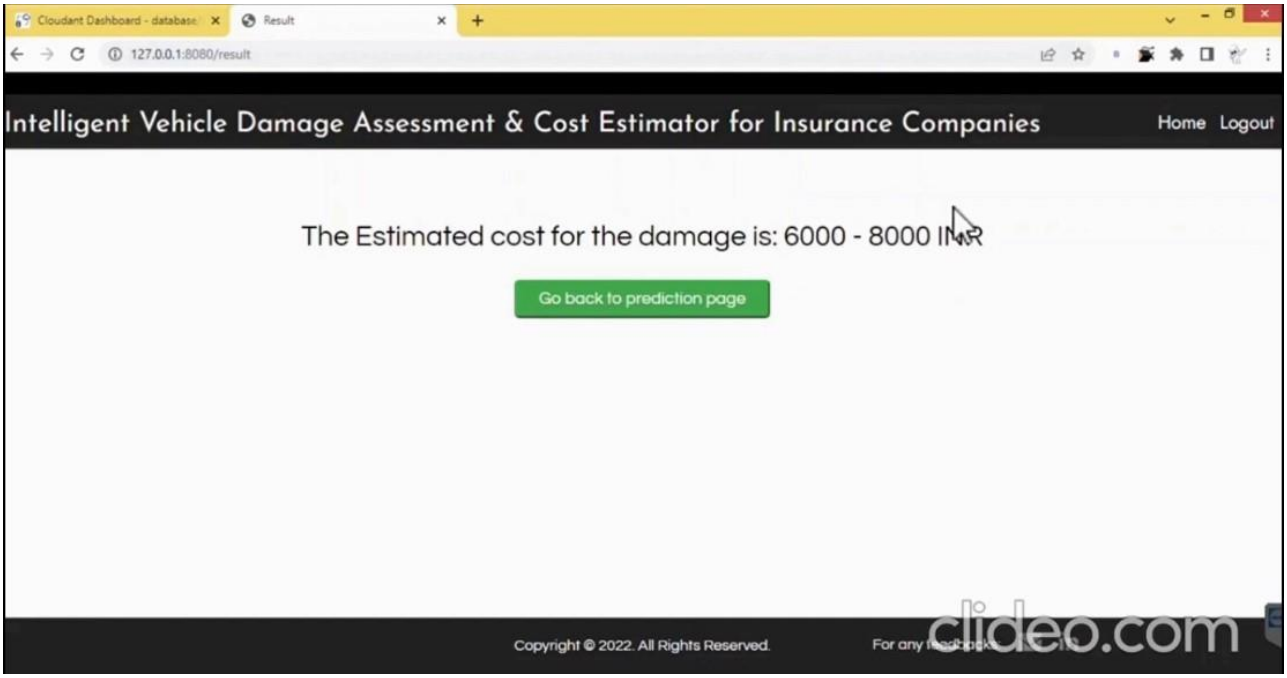
8.2. USER ACCEPTANCE TESTING



The registeration web page is tested with the already registered user information and hence it shows a message "You are already a member" by which the repeation of user infromation at database is prevented.



The login web page is tested with the invalid user information to check the invalid login testing into the webpage.



The prediction page is given with the test image of a damaged car to check the accuracy of the models.

9. RESULTS

9.1 PERFORMANCE METRICS

The performance of the Cost estimator for insurance companies is tested and assessed with the latency check, which is run over the prediction page. The time taken to load the image and predict the cost based on the damages in the vehicle is checked. The results show that the web application took less than 10s to provide the estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance.

- Repair cost optimization, total loss and agreed value
- Quick assessment by phone – without the need for a visit by the professional inspector
- Overseeing the repair of the vehicle
- Establishing the monetary and residual value of vehicles
- Assistance in court
- Accident investigation to check all the data provided on the claim file
- Our reports and dataset are customized and adapted to your workflow, minimizing changes to your processes

The results show that the web application took less than 10s to provide the estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance

10. ADVANTAGES

1. The Advantage of having an Intelligent Cost Estimator based on the damages can save the time and resource of the user in automatically evaluating the images with the damages using the Deep Learning models trained with the various car images
2. **Finding a proper data set-** Training machine learning models requires a sufficient data set of relevant images. The more varied the images are, the better the model will be able classify images appropriately.

3. Preprocessing image data sets is a crucial step in speeding up and obtaining better training results for models. This activity may span a variety of tasks: applying filters, removing noise, enhancing contrast, down sampling images, etc.

4. **Building a model-** After you have a quality data set at hand, there are still some considerations when building a machine learning model.

DISADVANTAGES

5. The Disadvantage of the project is expensive coding and time to develop the front end and back end of the web application
6. Creating and training a model takes time
7. **Optimizing performance and costs -** As insurance companies have to deal with damage assessment on a daily basis, the working solution also needs to demonstrate resonating performance

11. CONCLUSION

We conclude by suggesting this web application for damage assessment and cost estimation for the insurance companies. The web application is supported by the Deep Learning and IBM Watson cloud which stands for the complex image prediction and user information storage. The web application takes the user registration and login, The user can login into the prediction page using their ID and password. The prediction

takes the image input and the model can predict the input based on the perviour knowledge about the damages.

In future, The User Interface of the web application can be improved by updating the HTML and CSS codings. The improvement in UI can gives the better user exprience in future, The model's accuracy over various images can increased by trainning with various damaged images. The Image processing methods can be improved to achive higher performance of the model in the future.

12. FUTURE SCOPE

In future, The User Interface of the web application can be improved by updating the HTML and CSS codings. The improvement in UI can gives the better user exprience in future, The model's accuracy over various images can increased by trainning with various damaged images. The Image processing methods can be improved to achive higher performance of the model in the future.

13. APPENDIX

Github Repo:

https://github.com/IBM-EPBL/IBM-Project-9265-1658989879

VideoLink:

https://drive.google.com/drive/folders/1c1k5nvcbQPMOY8q9R4vYA4VdE-c4w7z7?usp=share_link

App.py

```
import re
import numpy as np
import os
from flask import Flask, app, request, render_template
from keras import models
from keras.models import load_model
from keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from keras.applications.inception_v3 import preprocess_input
import requests
from flask import Flask, request, render_template, redirect,
url_for
from cloudant.client import Cloudant

#Create Database
client = Cloudant.iam('00cba18f-2150-4961-9102-f29b9aee35de-
bluemix','ht_ByiEjrGeaitIZJTC-ri5_8Oq-dxTNHLGho1mpt0d5',
connect=True)
my_database = client.create_database('my_database')

#Loading the Model
```

```
model1 = load_model('Model/level.h5')
model2 = load_model('Model/body.h5')

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/index.html')
def home():
    return render_template('index.html')

@app.route('/register.html')
def register():
    return render_template('register.html')

@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
        '_id': x[1],
        'name': x[0],
        'psw': x[2]
    }
    print(data)

    query = {'_id': {'$eq': 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)
        response = request.get(url)
        return render_template('login.html', pred="Registration
```

```
Successful, Please login using your details")
    else:
        return render_template('register.html', pred="You are
already a member, Please login using your details")

@app.route('/login.html')
def login():
    return render_template('login.html')

@app.route('/afterlogin', methods=['POST'])
def afterlogin():
    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('login.html', pred="The Username
is not found")
    else:
        if((user==docs[0][0]['_id'] and
passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')

@app.route('/logout.html')
def logout():
    return render_template('logout.html')

@app.route('/prediction.html')
def prediction():
    return render_template('prediction.html')
```

```
@app.route('/result')
def res():
    if request.methods=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(256,256))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)

        img_data=preprocess_input(x)
        prediction1=np.argmax(model1.predict(img_data))
        prediction2=np.argmax(model2.predict(img_data))

        index1=['front','rear','side']
        index2=['minor','moderate','severe']

        result1 = index1[prediction1]
        result2 = index2[prediction2]
        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('prediction.html',prediction=value)

if __name__=="__main__":
    app.run(debug = False,port = 8080)
```