# INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

**ABSTRACT**

The motor insurance sector loses a lot of money as a result of leakage claims. The gap between the amount actually paid for claims and the amount that would have been paid had all of the best practises in the industry been followed is known as underwriting leakage. These results have been reached using both testing and visual assessment.

However, they do delay the processing of claims. 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 method for automatically locating the damaged area by comparing photos of the automobile from before and after an accident. This project's proposed a CNN model that can recognise a car's damage area. If users

upload images, the model can evaluate damage (be it a dent or scratch from an object), and it can also estimate the extent of damage. Insurance ﬁrms can handle claims more eﬃciently as a result. When accepting a car loan, particularly one for a used vehicle, lenders may also consider this model.

# INTRODUCTION

**PROJECT OVERVIEW**

Vehicles are signiﬁcantly rising in today's globe. Because there are more cars on the road, accidents happen more frequently because individuals are driving them at high speeds. When an accident occurs, the people ﬁle a claim with their auto insurance for the necessary funds to repair the car, because to inaccurate claims, the corporation

behaves improperly and doesn’t make payments now. This occurs as a result of claims leakage, which is the discrepancy between the sums secured by the ﬁrm and the sums that it should have secured in accordance with the claims. Even if the car's damage is

easily seen, the claim procedure will take longer than usual in accordance with company policy.

Despite the company's best efforts, there is a delay in the claims procedure.

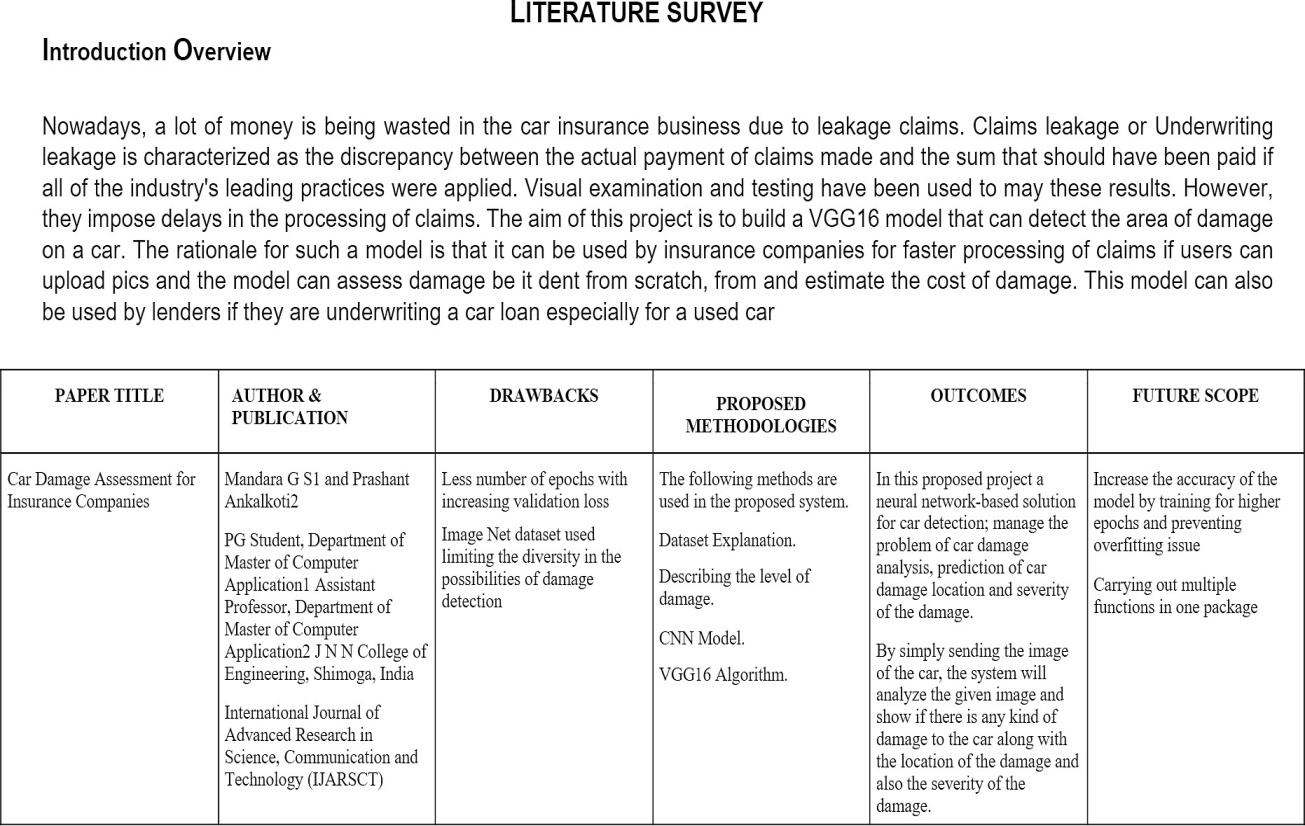
Differentiate the suggested approach to perhaps speed up the process of assessing automotive damage. Instead of taking hours to accomplish automotive damage detection if it were visually inspected, a system may perform it in a minute by just

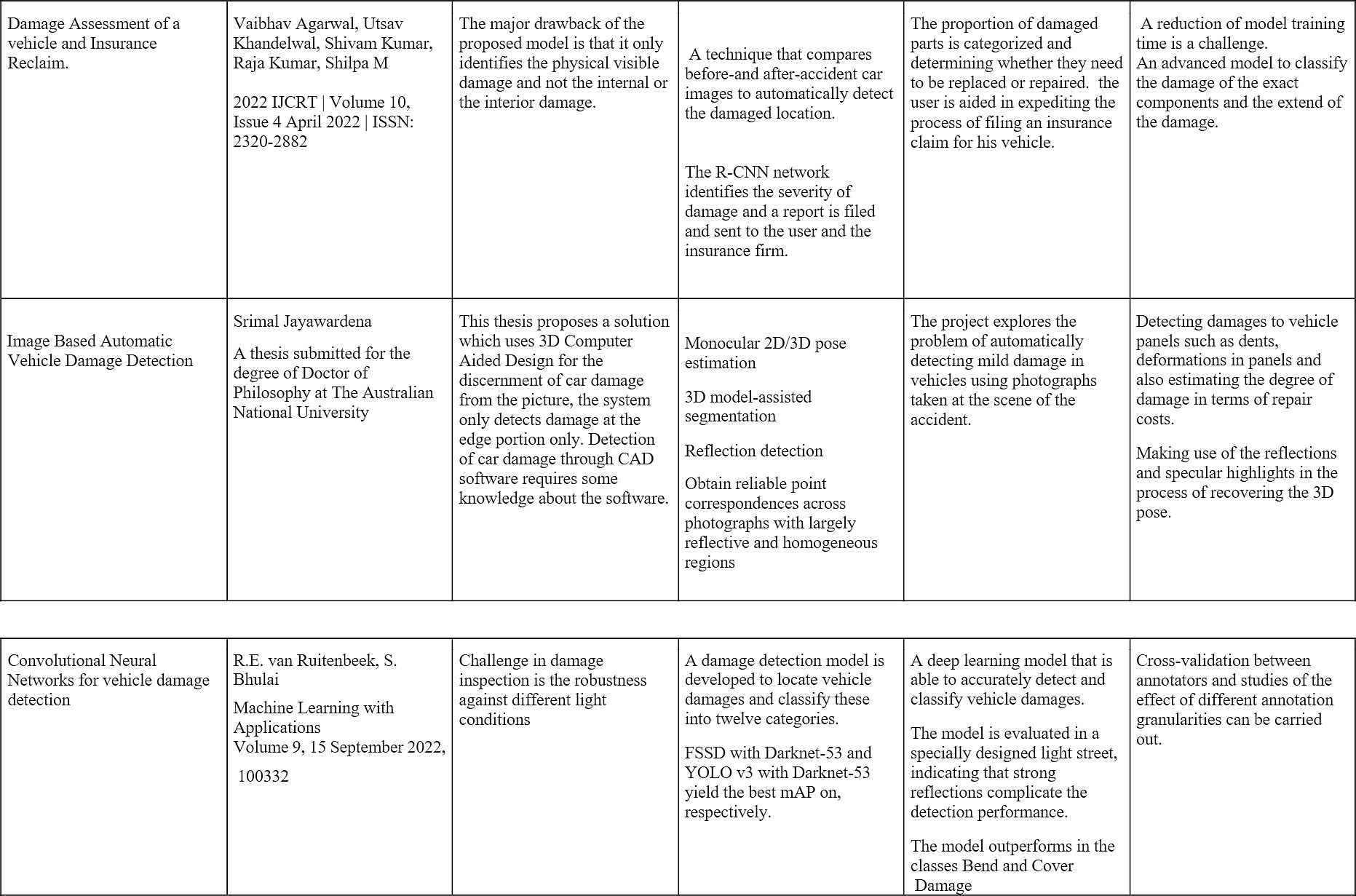
providing a picture of a damaged vehicle. The system can determine the analysis of the damage, the position of the damage, and the degree of the damage using machine learning and computer vision.

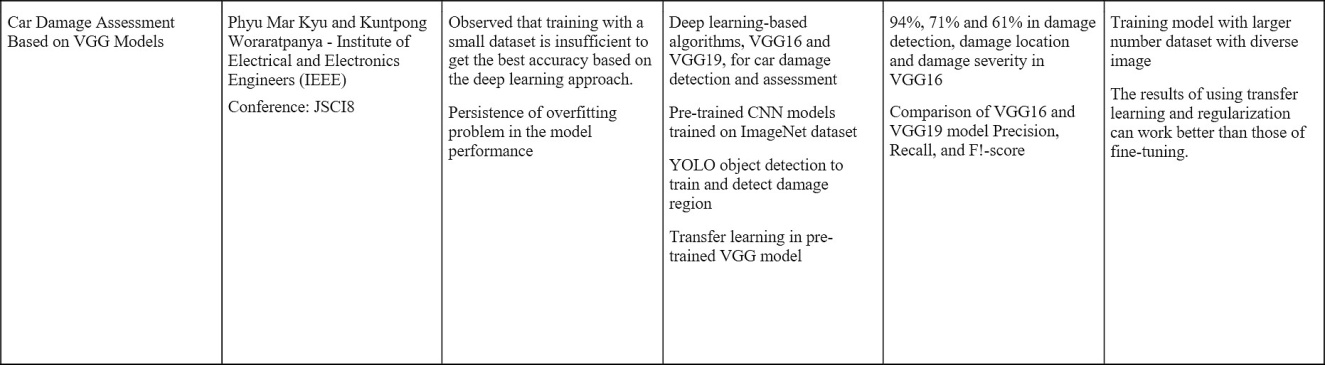
# PURPOSE

Today's world is seeing a substantial increase in automobiles. Because there are more automobiles on the road and more people are driving them at high speeds, accidents happen more frequently. When an accident happens, the parties involved submit a claim with their auto insurance to obtain the money needed to repair the vehicle since, according to false claims, the company acts inappropriately and withholds payments.

**LITERATURE REVIEW**







# EXISTING PROBLEM

* + 1. TITLE: Convolutional Neural Networks for vehicle damage detection, 2021 AUTHOR NAME: R.E. Ruitenbeek

Vehicle damage is becoming an increasing liability for shared mobility

providers. The high number of driver handovers necessitates the use of an accurate and quick inspection system capable of detecting minor damage and categorising it. To address this, a damage detection model is created that locates vehicle damages and categorises them into twelve groups. To improve detection performance, multiple deep learning algorithms are used, and the effect of various transfer learning and training strategies is evaluated. The ﬁnal model, which was trained on over 10,000 damage

photos, can detect minor defects in a variety of environments, including water and dirt. A performance evaluation using domain experts reveals that the model performs comparably. Furthermore, the model is tested in a specially designed light street, demonstrating how strong reﬂections complicate detection performance.

* + 1. TITLE: Deep Learning Based Car Damage Detection, Classiﬁcation and Severity AUTHOR NAME: Ritik Gandhi1, 2021

Because it is a manual procedure, resolving a claim in the accident insurance sector takes time, and there is a gap between the ideal and real settlement. We are using deep learning models to not only speed up the process, but also to deliver better customer service and boost insurance company proﬁtability. In this paper, we use multiple pre trained models such as VGG 16, VGG 19, Resnet50, and DENSENET to choose the top performing models. We ﬁrst use the Resnet50 model to determine whether or not the automobile is damaged, and if it is, we utilise the WPOD-net model to identify the licence plate. The YOLO model is used to detect the affected region. Finally, the damage severity is implemented using the DENSENET model. We discovered that

transfer learning outperforms ﬁne-tuning after applying multiple models. Furthermore, we present a framework that incorporates all of this into a single application, assisting in the automation of the insurance sector.

* + 1. TITLE: Car Damage Assessment to Automate Insurance Claim, 2022 AUTHOR NAME: Siddhant Gole

Car damage inspection is an essential stage in claim sanctioning, and the procedure is frequently delayed and erroneous, resulting in claim leakages. Our task is to create a web application connected with a deep learning model that receives user input in the form of photographs of damaged automobiles and assesses the damage to provide a cost report that the ﬁrm can use to approve the ﬁrst reimbursement. To detect and localise the damaged regions, the model employs the MASK R-CNN algorithm in conjunction with Faster RCNN. The device also includes a security module that detects and stores the vehicle's licence plate, body type, and logo data for veriﬁcation. Our goal is to develop a system that can detect damaged parts of a car using images and

generate a cost analysis report that the company can use to sanction the insurance amount. The task would be to develop an end-to-end system for detecting and classifying types of damage via images, as well as to implement a car number plate, body type, and logo detection system to verify car details.

* + 1. TITLE: Using Machine Learning Models To Compare Various Resampling Methods In Predicting Insurance Fraud, 2021

AUTHOR NAME: Ruixing Ming

Insurance fraud is one of the most prevalent kinds of fraud. In particular, the cost of automotive insurance fraud is signiﬁcant for property insurance companies and has a long-term inﬂuence on insurance businesses' pricing tactics. And Car insurance fraud detection has become required in order to reduce insurance prices. Although predictive models for detecting insurance fraud are widely used in practise, there are few

published research on the use of machine learning algorithms to identify insurance fraud, most likely due to a lack of available data. Evaluate 13 machine learning

approaches in this paper using real-world data. Predicting insurance fraud has become a big diﬃculty due to the uneven datasets in this domain. Because our data consists

primarily of "non-fraud claims" with a minor number of "fraud claims." As a result, classiﬁcation models predict fraud poorly; thus, the current study seeks to propose an approach that improves machine learning algorithms' results by using re-sampling

techniques, such as Random over Sampler, Random under Sampler, and hybrid methods, to address the issue of unbalanced data.

* + 1. TITLE: Evaluation of deep learning algorithms for semantic segmentation of car parts, 2021

AUTHOR NAME: Kitsuchart Pasupa

One of the most signiﬁcant operations in the auto insurance industry is the evaluation of accident-damaged vehicles. Currently, each fundamental component must be manually examined. It is believed that in the future, a smart device will be able to do this evaluation more effectively. We analysed and compared ﬁve deep learning algorithms for semantic segmentation of automobile parts in this work. Mask R-CNN served as the baseline reference method, while the other algorithms were HTC, CBNet, PANet, and GCNet. These ﬁve algorithms were used to do instance segmentation runs. HTC's ResNet-50 algorithm was the best for segmentation on various types of cars such as sedans, trucks, and SUVs. On our initial data set, it attained a mean average accuracy of 55.2 when distinct labels were allocated to the left and right sides, and 59.1 when a single label was assigned to both sides. Furthermore,

the models from each method were veriﬁed for robustness by running them on photos of components in a real-world setting with varying weather conditions such as snow,

frost, fog, and different lighting situations. When left and right sides were assigned different labels, GCNet achieved a mean performance under corruption, mPC = 35.2, and a relative degradation of performance on corrupted data, compared to clean data (rPC), of 64.4%, and mPC = 38.1 and rPC = 69.6% when left and right sides were considered the same part.

# REFERENCES

[1]. R.E. Ruitenbeek, Convolutional Neural Networks for vehicle damage detection, 2021

[2]. Ritik Gandhi1Deep Learning Based Car Damage Detection, Classiﬁcation and Severity, 2021

[3]. Siddhant Gole, Car Damage Assessment to Automate Insurance Claim, 2022

[4]. Ruixing Ming, Using Machine Learning Models To Compare Various Resampling Methods In Predicting Insurance Fraud, 2021

[5]. Kitsuchart Pasupa, Evaluation of deep learning algorithms for semantic segmentation of car parts, 2021

# PROBLEM STATEMENT DEFINITION

In existing system, the procedure of making an insurance claim for an automobile is laborious, and there is a delay before the ﬁrst reimbursement is authorised. Insurance

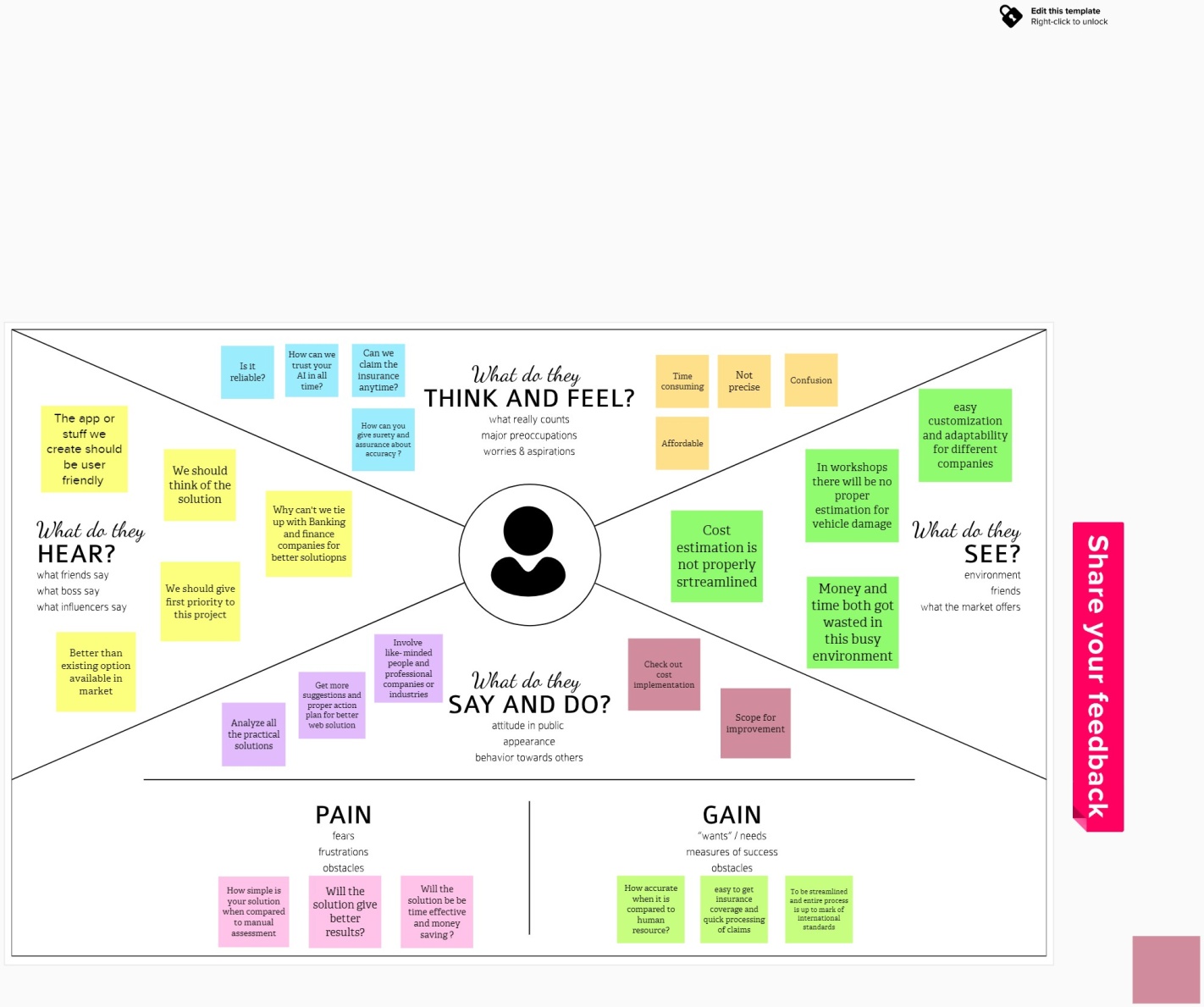
ﬁrms lose millions of dollars each year due to claim leakage as a result of the expansion of the vehicle sector and the daily rise in the number of accidents. The discrepancy

between the company's actual spending and what they should have really spent is known as claim leakage. Ineffective claim processing, erroneous payments, human

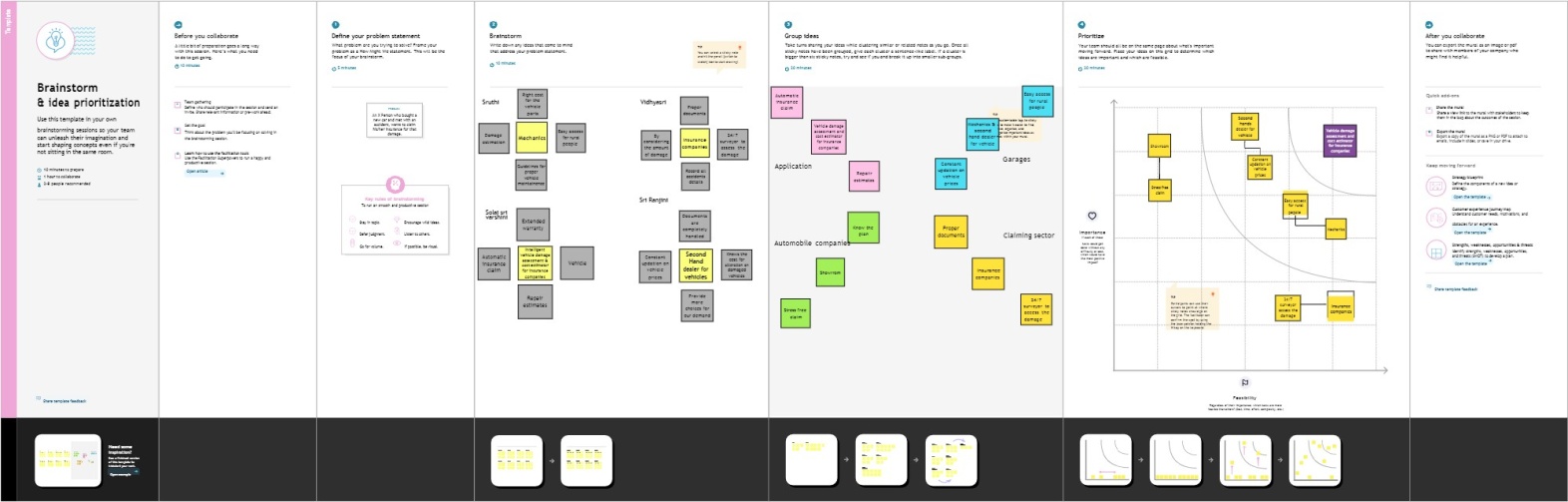
error such as a lack of quality control or poor customer service or even claim fraud may be to blame for this. Auditing closed claim ﬁles is the only way to ﬁnd claim leakage.

# IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



IDEATION & BRAINSTORMING



# PROPOSED SOLUTION

The proposed approach collects photographs of a person's damaged automobile, then utilises those images as input for a deep learning model that use image processing to recognise the elements of the image and determine the percentage of the vehicles’’ damage. After then, the images are separated into two groups: replace and repair. When the damage percentage is less than 80, the damaged part must be replaced; however, in the other case, the compensation amount is set depending on the damage percentage.

Finally, it generates a comprehensive analysis report on the vehicle that is used to ask the insurance company for payment.

# PROBLEM SOLUTION FIT

There is no systematic approach to receive a rapid answer from an insurance company. A week of waiting is required. The proposed solution should enable consumers to contact with the insurance provider and receive payments both online and oﬄine. After uploading the damaged image and determining the extent of the damage,

the user may obtain insurance only if the company approves the damaged image and the condition is more than 80%.

# REQUIREMENT ANALYSIS

**FUNCTIONAL REQUIREMENT**

# Framework Creation:

This approach provides a way for evaluating vehicle damage that insurance companies may utilise when processing claims. This module offered a framework for submitting a vehicle's damaged parts and requesting insurance from an organisation. The dataset needed to train the Damage Detection and it has prepared by an admin. In order to make the images useful for training, they were manually annotated; damages were categorised into 7 distinct types such as Door Dent, Bumper Dent, Body Scratch, Broken Windshield, Broken Glass, Broken Lights and Smash By modifying its settings and loading the learned dataset, the model was set up to train on user data.

# Object Detection:

Employ a specially trained CNN model utilising transfer learning on to identify the object. This model takes different forms of damage into account validation sets such as Bumper Dent, Bumper Scratch, Door Dent, Door Scratch, Glass Shattered, Head Lamp, Tail Lamp, Undamaged, etc. The classiﬁcation of car damage severity is as follows:

Minor Damage which typically involves slight damage to the vehicle that does not impede the vehicle to cause severe injuries. It includes the headlight scratches, dents and digs in the hood or windshield, from gravel or debris, scratches in the paint.

Moderate Damage which deals with any kind of damage that impairs the functionality of the vehicle in any way is moderate damage. It involves large dents in hood, fender or door of a car. Even if the airbags are deployed during collision, then it comes under moderate damage. Severe Damage – Structural damages such as bent or twisted

frames, broken/bent axels, and missing pieces of the vehicles and in some cases even the destruction of airbags. These types of damages are a big threat to the human life.

# Damage Detection:

To locate damused which combines object localisation and classiﬁcation to provide a bounding box and a class for each item for object detection. Use CNN to generate a

convolutional features map from an image to forecast the class and bounding box of an item. If the car is undamaged then it simply detects it and if it’s a damaged one, then

there are further localizations made models. The model shows accuracy on the validation set. To automate such a system, the easiest method would be to build a

Convolution Neural Network model capable of accepting images from the user and determining the location and severity of the damage. The model is required to pass

through multiple checks would ﬁrst ensure that given image is that of a car and then to ensure that it is in fact damaged. These are the gate checks before the analysis begins. Once all the gate checks have been validated, the damage check will commence. The model will predict the location of the damage as in front, side or rear, and the severity of such damage as in minor, moderate or severe.

# Claim Insurance:

The procedure of claiming insurance is done by persons who are in need. For access to the company's insurance, the user must register and authenticate. After that, users may access their insurance information and submit an insurance claim request. The request for an insurance claim can be viewed and approved by the insurance company. Once the damaged image has been uploaded and the degree of the damage has been determined, the user may receive insurance only if the ﬁrm accepts the damaged image and the condition is greater than 80%.

# NON FUNCTIONAL REQUIREMENTS

**Usability**

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

# Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

# Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

# Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisﬁed.

# Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

# Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

# PROJECT DESIGN

**DATA FLOW DIAGRAMS**

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identiﬁes each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data ﬂow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design

teams visualize how data is processed and identify or improve certain aspects.

# LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data ﬂows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identiﬁes internal data stores that must be present in order for the system to do its job, and shows the ﬂow of data between the various parts of the system.

.

# LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using

between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.

# LEVEL 2

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD deﬁnes each domain boundary and illustrates the logical movement and transformation of data within the deﬁned

boundary. The diagram shows 'what' input data enters the domain, 'what' logical

processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modelling and one of the oldest.

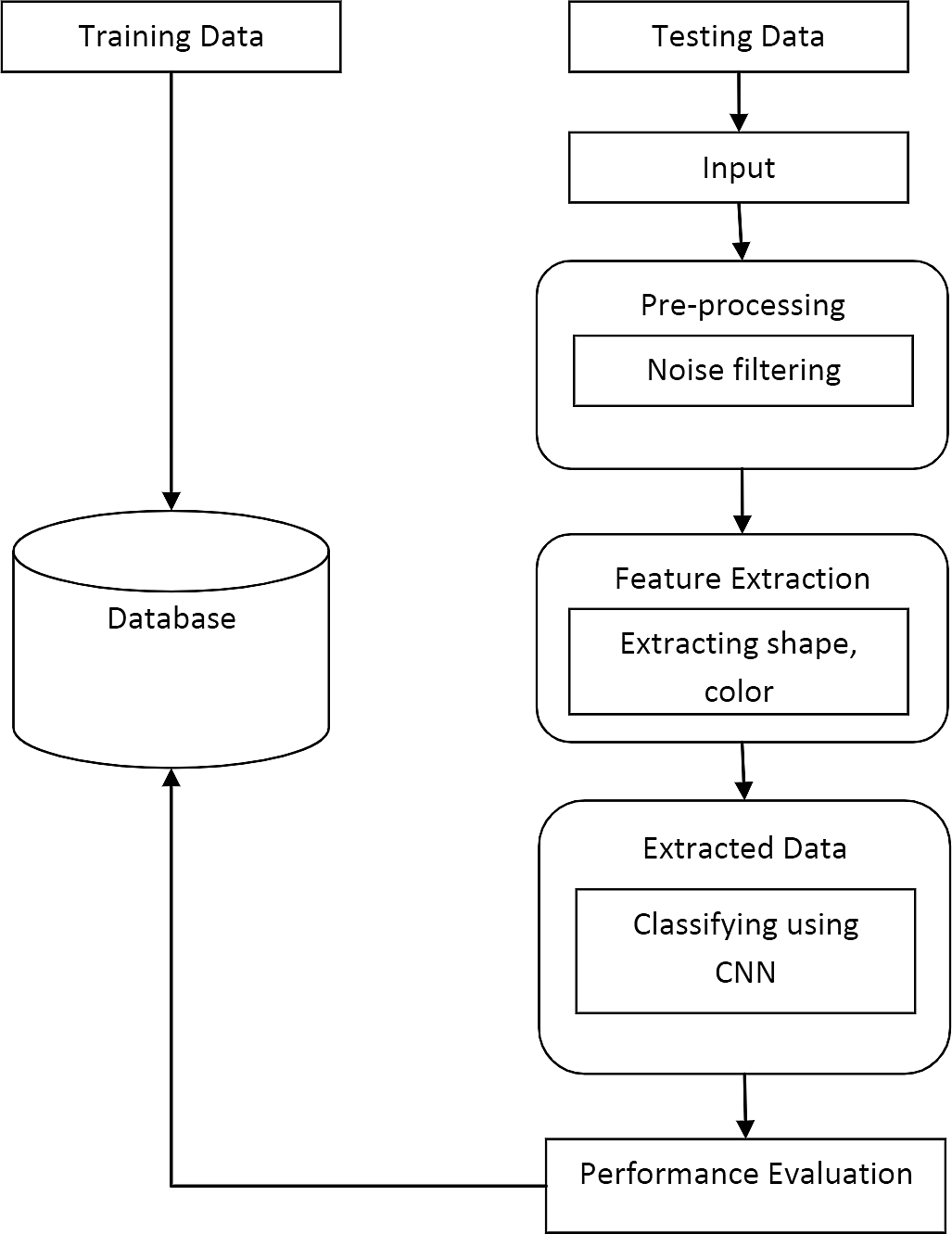
# SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that deﬁnes the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the

relationships (e.g. the behavior) between them.

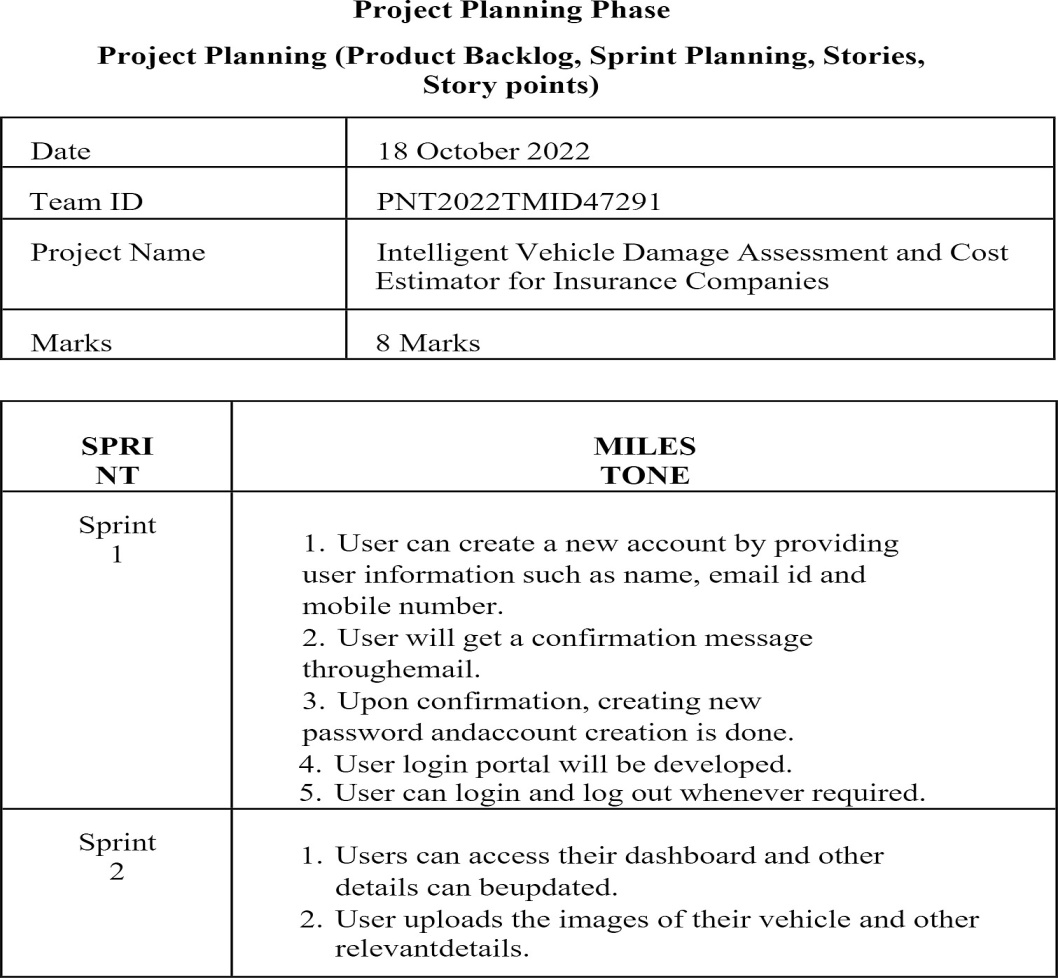
It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to

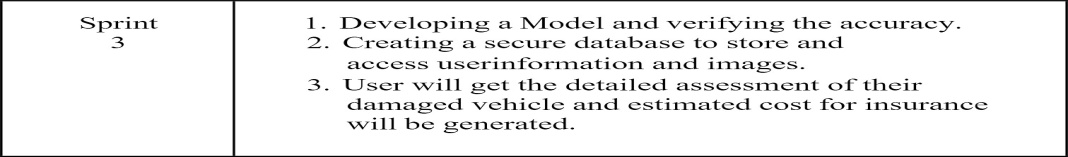
formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).



# USER STORIES

**PROJECT PLANNING & SCHEDULING**





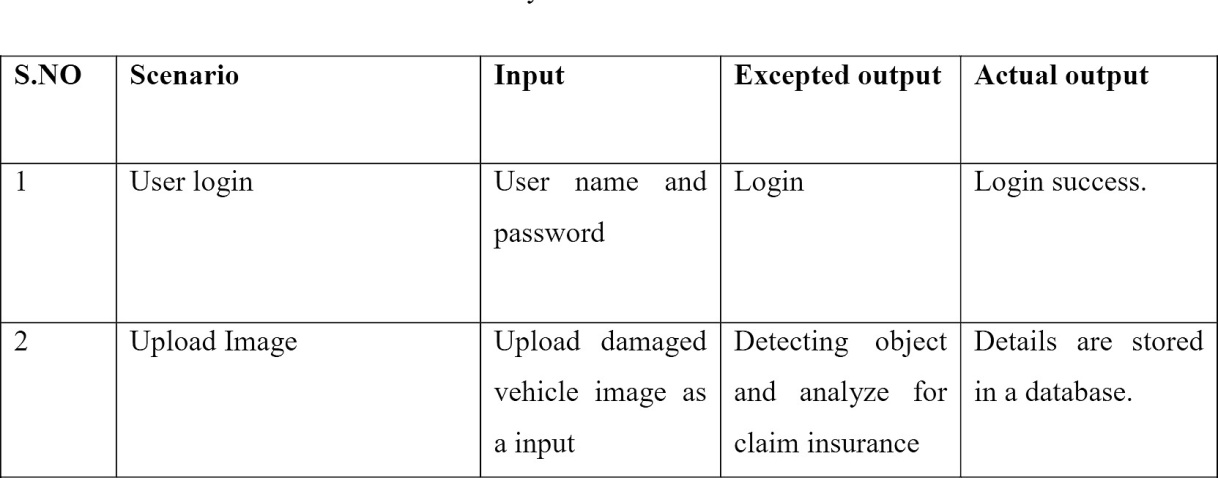
# TESTING

**TEST CASE**

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.

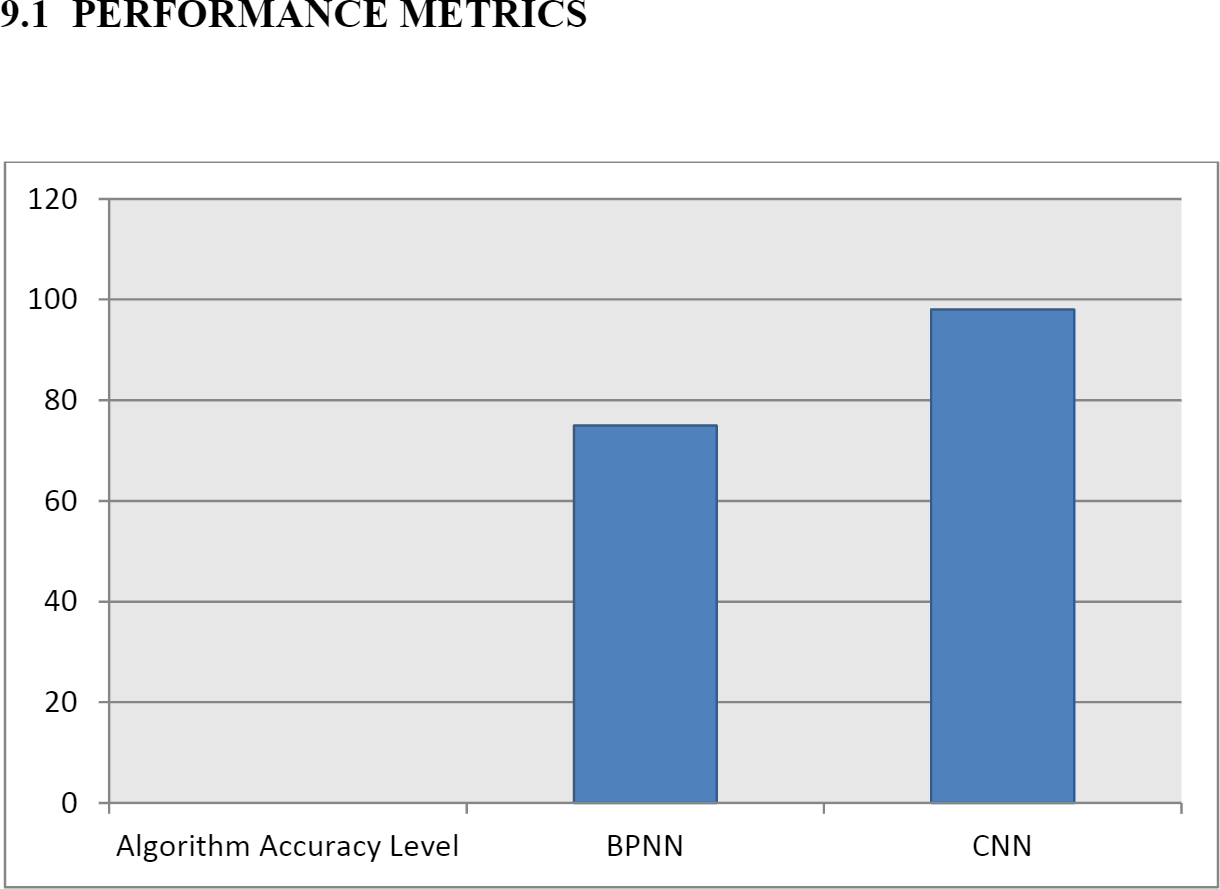


# USER ACCEPTANCE TESTING

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).

# RESULTS



**ADVANTAGES & 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

# 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.

# 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.

# 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"

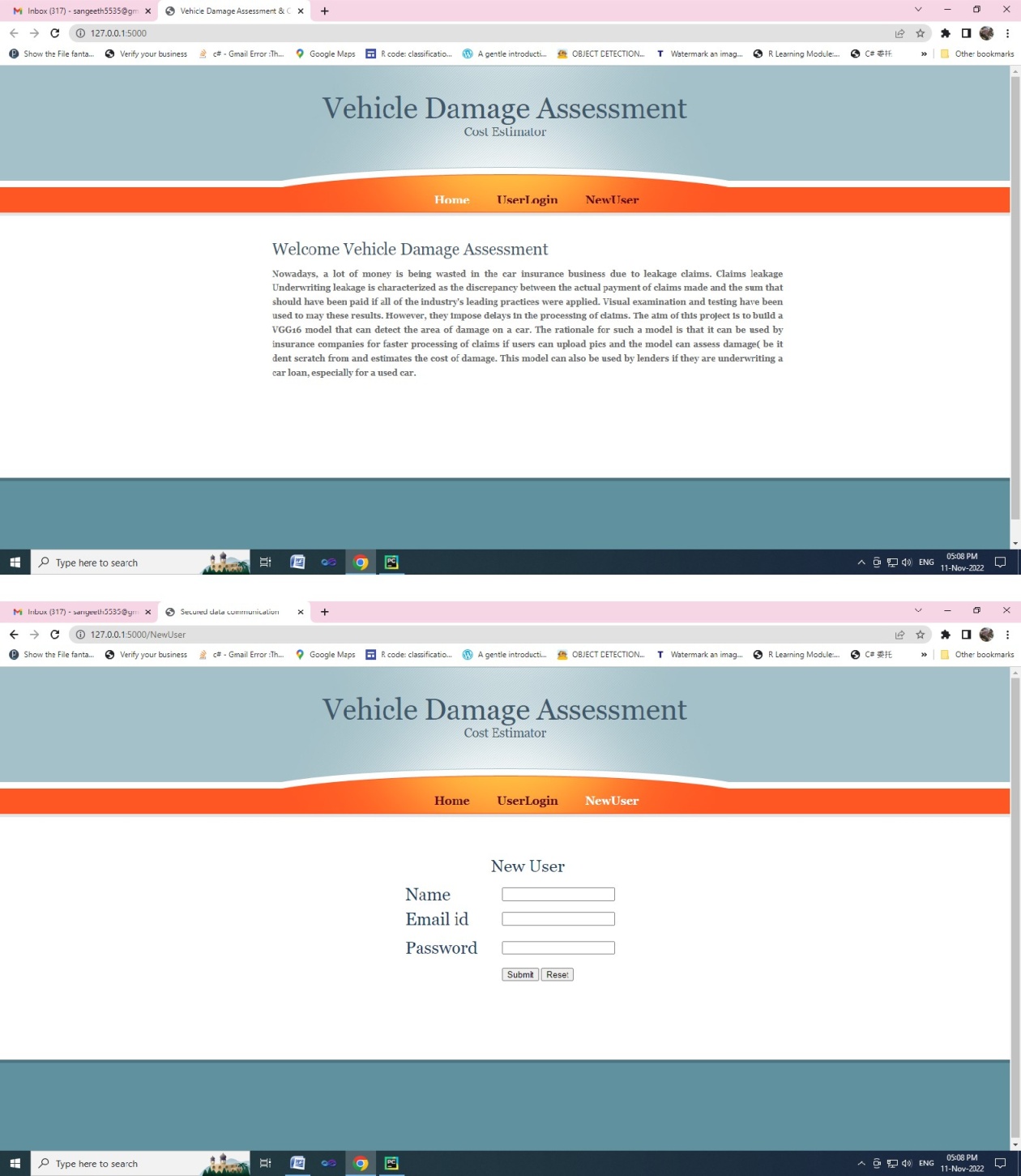
elif (result1 == "side" and result2 == "severe"): value = "12000 - 15000 INR"

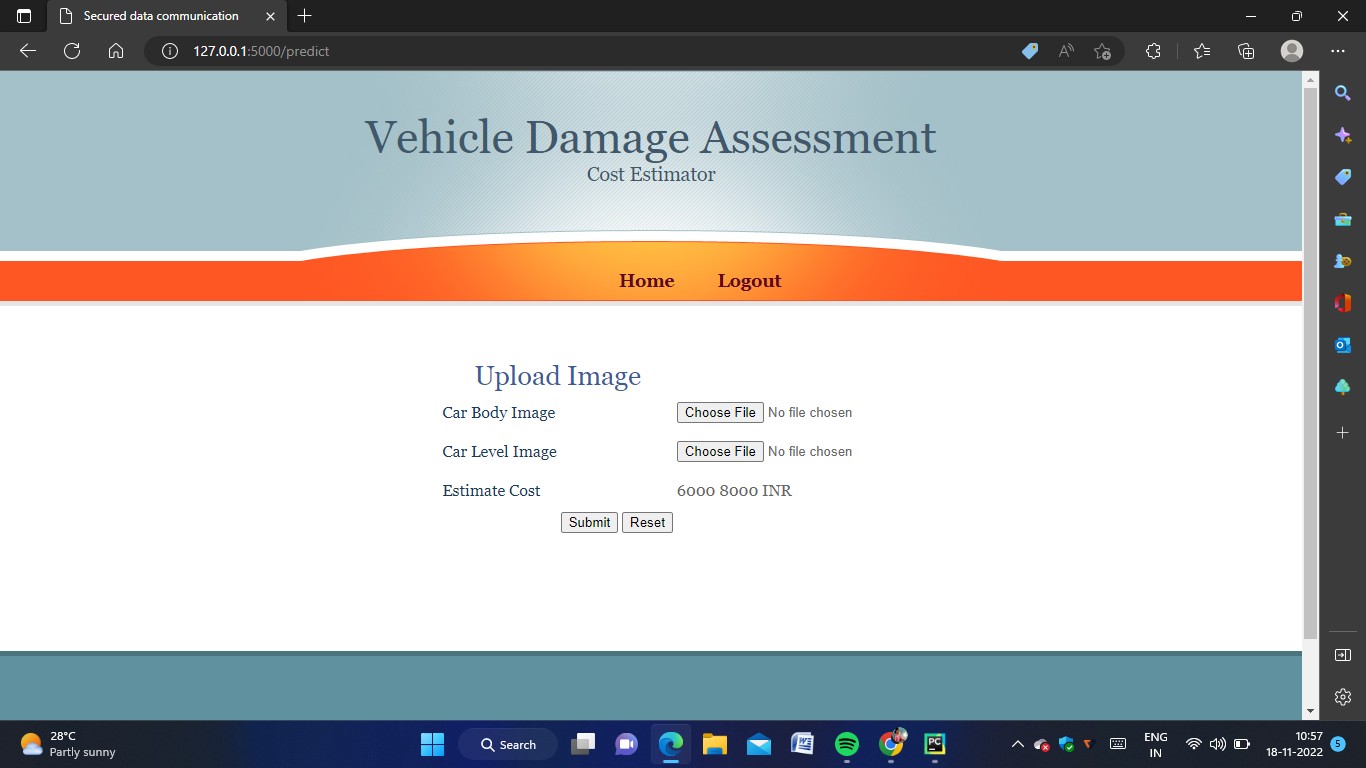
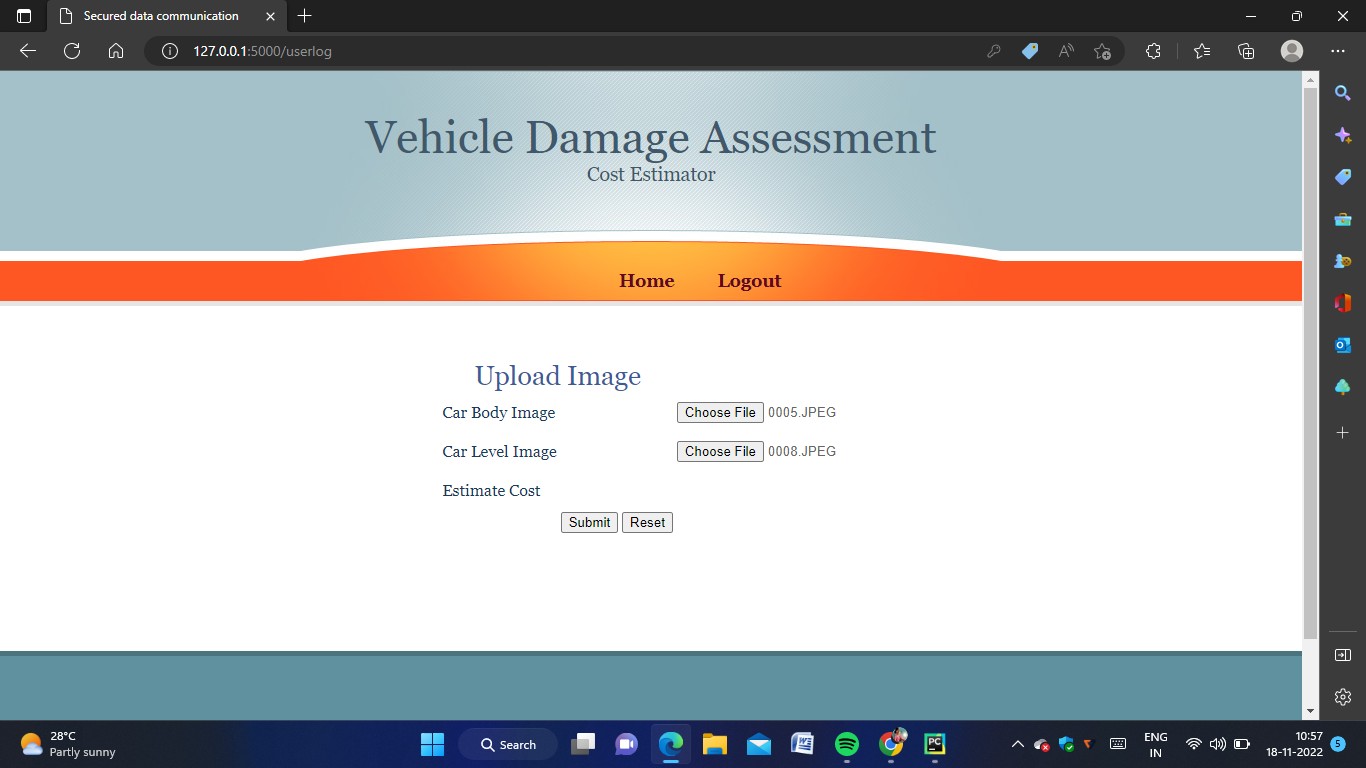
else:

value = "16000 - 50000 INR"

return render\_template('userhome.html', prediction=value)

if name == ' main ': app.run(debug=True, use\_reloader=True)





**GITHUB DEMO LINK**

https://github.com/IBM-EPBL/IBM-Project-48861-1660813780/tree/main/FINAL%20DELIVARABELS/PROJECT%20DEMONSTRATION%20VIDEO