ABSTRACT

The motor insurance sector loses a lot of money as a resultof 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 proposeda 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 claimsmore efficiently as a result. When accepting a car loan, particularly one for a used vehicle, lenders may also consider this model.

TABLE OF FIGURES

Performance Metrics

ADVANTAGES & DISADVANTAGES

CONCLUSION

FUTURE SCOPE

APPENDIX

Source Code GitHub& Project Demo Link

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 sumsthat 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 degreeof the damage using machinelearning 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 obtainthe money neededto repair the vehicle since,according to false claims, the company acts inappropriately and withholds payments.

LITERATURE SURVEY

TITLE: Convolutional NeuralNetworks for vehicledamage 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.

TITLE: Deep LearningBased Car Damage Detection, Classification and Severity AUTHOR NAME: Ritik Gandhi1, 2021

Because it is a manual procedure, resolving a claimin 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 damageseverity is implemented using the DENSENETmodel. 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.

TITLE: Car DamageAssessment to AutomateInsurance 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 damagedautomobiles and assessesthe damage to provide a cost reportthat the firm can use to approve

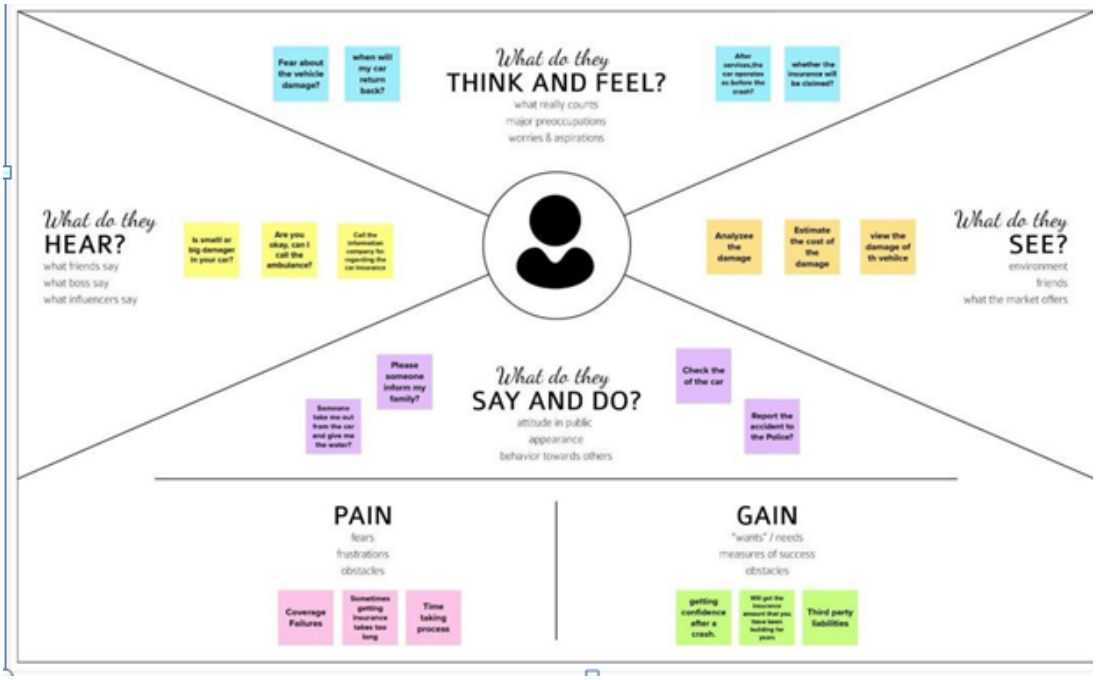
the first reimbursement. To detect and localise the damaged regions, the model employs the MASK R-CNN algorithm in conjunction with Faster RCNN. The device also includes a security module that detects and stores the vehicle's licence plate, body type, and logo data for verification. Our goal is to developa system that can detectdamaged parts of a car using images and generate a cost analysisreport 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 damagevia images, as well as to implement a car number plate,body type, and logo detection system to verify car details.

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 knownas 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 & PROPOSEDSOLUTION

EMPATHY MAP CANVAS



IDEATION & BRAINSTORMING

PROPOSED SOLUTION

The proposed approach collects photographs of a person's damaged automobile, then utilises thoseimages as input for a deep learningmodel that use image processing to recognise the elements of the image and determine the percentage of the vehicles'' damage. Finally, it generates a comprehensive analysis report on the vehicle that is used to ask the insurance company for payment.

Project Design Phase-IProposed Solution Template

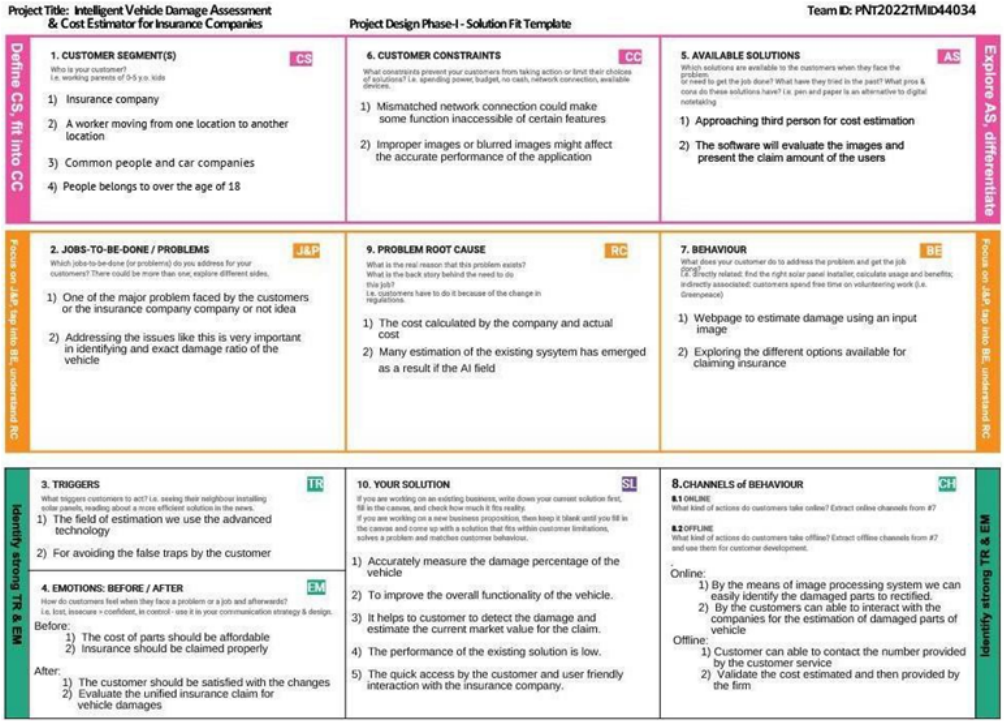
Project team shall fill the followinginformation in proposedsolution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To build a VGG16 model that can detect the area of damageon 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 modelcan 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 imagepreprocessing 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 ModelObject 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	1. AIbased car detection. 2. 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 basedon the severity of the damage.
5.	Business Model (Revenue Model)	Subscription and advertising model
6.	Scalability of the Solution	It allowsthe client to avoid giving the total amount of insurance to the policyholder for a small amount of damage.

PROBLEM SOLUTION FIT

There is no systematic approach to receivea 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 insuranceonly if the company approves the damaged image and the condition is more than 80%.



REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

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 manuallyannotated; damages were categorised into 7 distincttypes such as Door Dent, Bumper Dent, Body Scratch, Broken Windshield, Broken Glass, Broken Lights and Smash By modifying its settingsand 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 damageseverity is as follows: Minor Damage which typically involves slight damageto the vehicle that does not impede the vehicleto 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 airbagsare 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 damagesare a big threat to the human life.

Damage Detection:

To locate damaged areas in a picture and create a bounding box around each object found, object localization is used 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. Ifthe car is undamaged

then it simplydetects it and if it’s a damagedone, then there are further localizations made models. The model shows accuracy on the validation set. To automatesuch 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 ofa car and then to ensure that it is in fact damaged. These are the gatechecks before the analysis begins. Once all the gate checks have been validated, the damage check will commence. The model will predict the locationof 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 theusers 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 measureof a system's abilityto 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

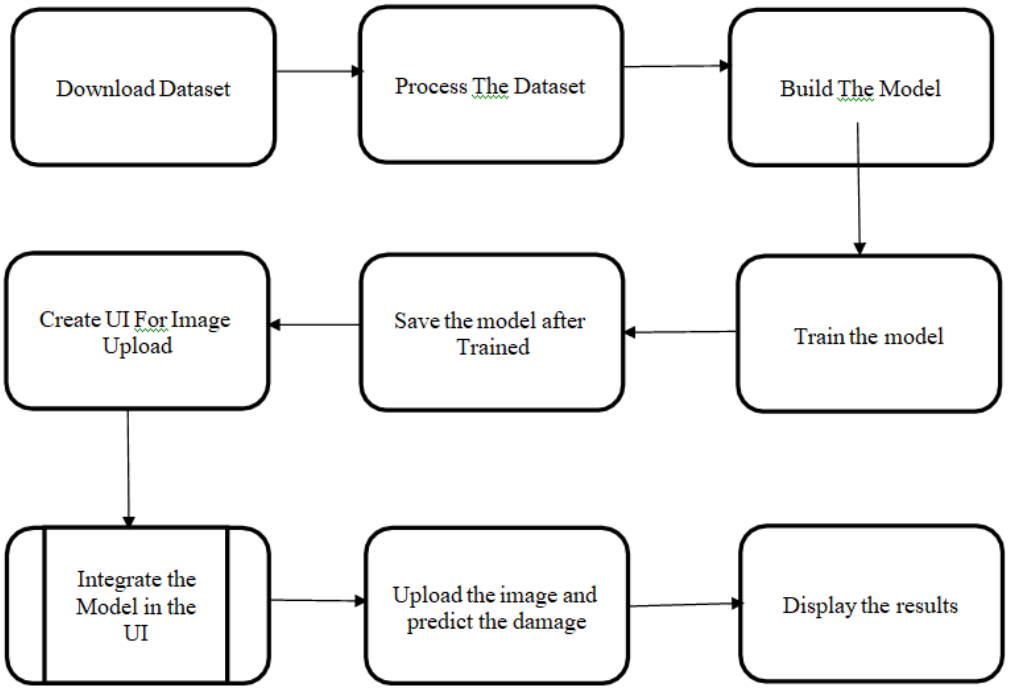
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

PROJECT DESIGN

DATA FLOW DIAGRAMS

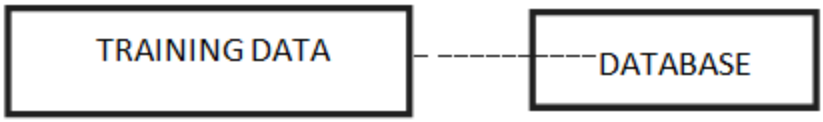
A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify externalinputs and outputs,determine how the inputs and outputs relateto 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.

DATA FLOW DIAGRAM:



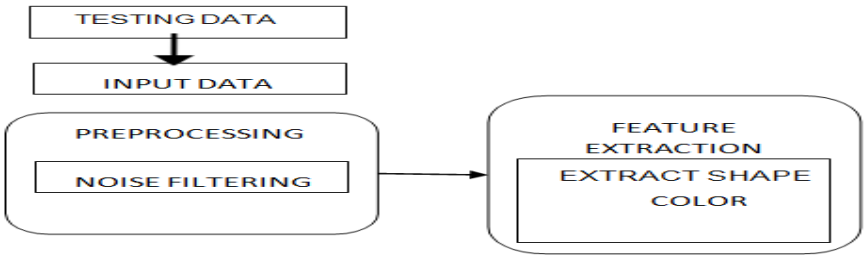
LEVEL 0

The Level 0 DFD shows how the systemis 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 systemas 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.

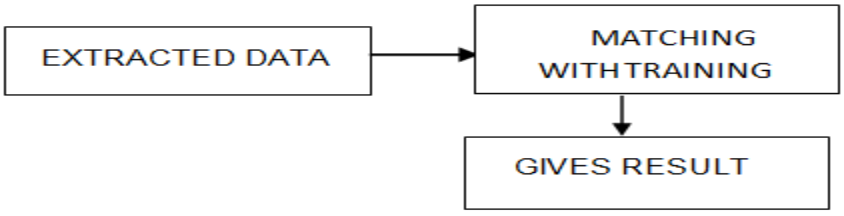


LEVEL 2

A Data FlowDiagram (DFD) tracksprocesses and their data paths within the business or system boundary under investigation. A DFD defineseach domain boundaryand 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).

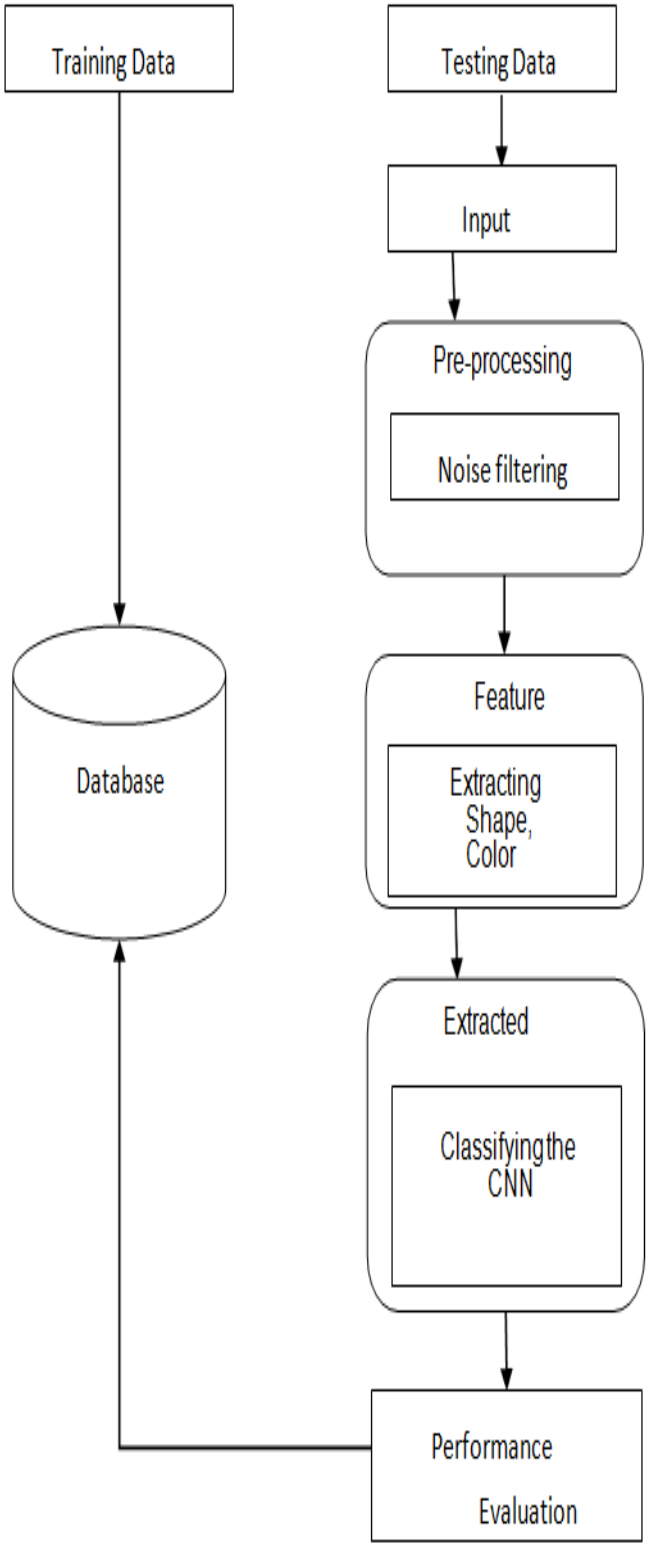


SOLUTION & TECHNICAL ARCHITECTURE

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).

SOLUTION ARCHITECTURE:

Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance


Training Data
Testing Data
Input
Pre-processing
Noise filtering
Database
Feature
Extracting
Shape,
Color
Extracted
Classifying the
CNN
Performance
Evaluation

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 the application by entering my email-Id and password and confirming the User Id and Password	I can access the dashboard/My account	High	Sprint-1
Customer details	login	USN-2	As a user, I will confirm that I have received my confirmation mail with my registered application	I canaccess the predicted page	High	Sprint-1
Customer Uses	Gmail access	USN-4	As a user, I can register the application through thegmail	I can sign up and access the details via the gmail	Medium	Sprint-2
Customer mandatory work	Upload the damaged image of the vehicle	USN-5	As a user, I can login into application with my own entering the email and password	I can scan thevehicle completely through camera	High	Sprint-3
Customer value	Cost is based on the damage of the vehicle	USN-6	Insurance would be provide based on the damage of the vehicle	I can get estimated insurance cost price	High	Sprint-3
Customer Care Executive	Effective Customer Support	USN-7	We can provide the excellent user assistance for the application for insurance	I can get theeffective customer support	Medium	Sprint-4

Administrator	To complete the customer work	USN-8	We will respond towards the customer needs in a goodway without mistakes	I canget effective customer support	High	Sprint-4

SPRINT PLANNING& ESTIMATION

Use the below template to create productbacklog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As an owner of a particular vehicle, I can log into the application by entering email & password.	2	High	vincent leo vIMAL
Sprint-1	User Confirmation	USN-2	As an owner of a particular Vehicle , I will receive confirmation email once I have registered for the application.	1	Medium	sanjeevi

Sprint-1	Login	USN-3	As an owner of a particular vehicle, I can log into the application by entering email & password.	2	High	navin
Sprint-2	Data Collection	USN-1	Download the dataset used in intelligent vehicle damage assessment & cost estimator for insurance companies.	2	High	romil
Sprint-2	Image Pre Processing	USN-1	Improve the image data that suppresses unwilling distortions or enhances some image features important for further processing, although	2	High	sanjeevi

SPRINT DELIVERYSCHEDULE

Sprint	Total Story Point	Duration	SprintStart Date	Sprint End Date (Planne- d)	Story Points (Com- pleted as on Planned end date)	Sprint release date (Actual)
Sprint-1	20	4 Days	24Oct 2022	30 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	28Oct 2022	03 Nov 2022	20	04 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	09 Nov 2022	20	11 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	18 Nov 2022	20	19 Nov 2022

a. CODING & SOLUTIONING

FEATURE 1

DASHBOARD

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>DASHBOARD</title>
    <link rel="stylesheet" href="style.css">
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1"><link
rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font
awesome/4.7.0/css/font-awesome.min.css">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<style>
*{
    margin: 0;
    padding: 0;
}
.main{
    width: 100%;
    background:linear-gradient(to top, rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%),
url("/static/images/2.jpg");
    background-position: center;
    background-size: cover;
    height: 100vh;
}
.navbar{
    width: 1200px;
    height: 75px;
```

```
    margin: auto;
}
.icon{
    width: 200px;
    float: left;
    height: 70px;
}
.logo{
    color: #ff7200;
    font-size: 35px;
    font-family: Arial;
    padding-left: 20px;
    float: left;
    padding-top: 2px;
    margin-top: 5px
}
.menu{
    width: 700px;
    float: right;
    height: 70px;
}
ul{
    float: right;
    display: flex;
    justify-content: center;
    align-items: center;
}
ul li{
    list-style: none;
    margin-left: 62px;
margin-top: 27px;
    font-size: 14px;
}
ul li a{
    text-decoration: none;
    color: #fff;
    font-family: Arial;
    font-weight: bold;
    transition: 0.4s ease-in-out;
```

```
}
ul li a:hover{
    color: #ff7200;
}
.search{
    width: 330px;
    float: left;
    margin-left: 270px;
}
.srch{
    font-family: 'Times New Roman';
    width: 200px;
    height: 40px;
    background: transparent;
    border: 1px solid #ff7200;
    margin-top: 13px;
color: #fff;
    border-right: none;
    font-size: 16px;
    float: left;
    padding: 10px;
    border-bottom-left-radius: 5px;
    border-top-left-radius: 5px;
}
.btn{
    width: 100px;
    height: 40px;
    background: #ff7200;
    border: 2px solid #ff7200;
    margin-top: 13px;
    color: #fff;
    font-size: 15px;
    border-bottom-right-radius: 5px;
    border-bottom-right-radius: 5px;
    transition: 0.2s ease;
    cursor: pointer;
}
.btn:hover{
    color: #000;
```

```
}
btn:focus{
    outline: none;
}
.srch:focus{
    outline: none;
}
.content{
    width: 1200px;
    height: auto;
    margin: auto;
    color: #fff;
    position: relative;
}
.content .par{
    padding-left: 20px;
    padding-bottom: 25px;
    font-family: Arial;
    letter-spacing: 1.2px;
    line-height: 30px;
}
.content h1{
    font-family: 'Times New Roman';
    font-size: 50px;
    padding-left: 20px;
    margin-top: 9%;
letter-spacing: 2px;
}
.content .cn{
    width: 160px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-bottom: 10px;
    margin-left: 20px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    transition: .4s ease;
```

```
}
.content .cn a{
    text-decoration: none;
    color: #000;
    transition: .3s ease;
}
.cn:hover{
    background-color: #fff;
}
.content span{
    color: #ff7200;
    font-size: 65px
}
.form{
    width: 300px;
    height: 340px;
    background: linear-gradient(to top, rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
    position: absolute;
    top: -20px;
    left: 870px;
    transform: translate(0%,-5%);
    border-radius: 10px;
    padding: 25px;
}
.form h2{
    width: 220px;
    font-family: sans-serif;
    text-align: center;
    color: #ff7200;
    font-size: 22px;
    background-color: #fff;
    border-radius: 10px;
    margin: 2px;
    padding: 8px;
}
.form input{
    width: 240px;
    height: 35px;
    background: transparent;
```

```
    border-bottom: 1px solid #ff7200;
    border-top: none;
    border-right: none;
    border-left: none;
    color: #fff;
    font-size: 15px;
    letter-spacing: 1px;
    margin-top: 30px;
    font-family: sans-serif;
}
.form input:focus{
    outline: none;
}
::placeholder{
    color: #fff;
    font-family: Arial;
}
.btnn{
    width: 240px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-top: 30px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    color: #fff;
    transition: 0.4s ease;
}
.btnn:hover{
    background: #fff;
    color: #ff7200;
}
.btnn a{
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
.form .link{
```

```
    font-family: Arial, Helvetica, sans-serif;
    font-size: 17px;
    padding-top: 20px;
    text-align: center;
}
.form .link a{
    text-decoration: none;
    color: #ff7200;
}
.liw{
    padding-top: 15px;
    padding-bottom: 10px;
    text-align: center;
}
.icons a{
    text-decoration: none;
    color: #fff;
}
.icons ion-icon{
    color: #fff;
    font-size: 30px;
    padding-left: 14px;
    padding-top: 5px;
    transition: 0.3s ease;
}
.icons ion-icon:hover{
    color: #ff7200;
}
.end {
            overflow: hidden;
            background-color: rgb(63, 63, 63);
            position: auto;
            bottom: 0;
            height: 55px;
            width: 100%;
        }
</style>
</head>
<body = onload="flashMessage()">
```

```
<script>
  function flashMessage(){
    if("{{flash_message}}" == "True"){
      alert("account created successfully")
    }
    if("{{flash_message}}" == "Fals"){
          alert("invalid credentials")
    }
    if("{{flash_message}}" == "Fal"){
      alert("Logged in successfully")
    }
  }
</script>
<body style:"background-image:static\images\2.jpg;">
<div class="main">
    <div class="navbar">
        <div class="icon">
            <h6 class="logo"></h6>
        </div>
        <div class="menu">
            <ul>
                <li><a href="{{ url_for('dashboard') }}">HOME</a></li>
                <!--<li><a href="{{ url_for('login') }}">LOGIN</a></li>-->
                <!--<li><a href="{{ url_for('register') }}">REGISTER</a></li>-->
                <li><a href="{{ url_for('prediction') }}">PREDICTION</a></li>
                <li><a href="{{ url_for('logout') }}">LOGOUT</a></li>
                <li><a href="#section -1">ABOUT</a></li>
            </ul>
        </div>
    </div>
    <div class="content">
        <h1>Intelligent Vehicle <br><span>Damage Assessment &<br>Cost Estimator for
</span><br>Insurance Companies</h1>
          <br><br><br>
            <button class="cn"><a href="{{ url_for('prediction')
}}">PREDICTION</a></button>
        <!--form action="dashboard" method="POST">
            <div class="form">
                <input type="email" name="email" id="email" placeholder="Enter Your Email Id">
```

```
                <input type="password" name="password" id="password" placeholder="Enter Your
Password ">
                <button class="btnn"><a href="">Login</a></button>
                <p class="link">Don't have an account<br>
                <a href="#">Sign up </a> here</a></p>
              </div>
                            </form!-->
                </div>
              </div>
        </div>
    </div>
<br><br><br><center>
          <div id= "section -1"> <h2 style:font-family:'Times New Roman', Times,
serif;"><center>ABOUT PROJECT</center></h2></div>
<br><br>
<p style:"font-size:50px;font-family: 'Times New Roman', Times, serif;">Vehicle Damage
Detection is used to reduce claims leakage during insurance processing.Vechile Inspectin and
validation are usually done.As it takes a long time,because a person need to come and Inspect
Damage.Here we are trying to Automate the procedure .Using this Automation we can avoid
time consumption for Insurance claim procedure</p>
<br><br><br><br>
 <a style:"color:white;
                padding: 22px;
                background-color: black;
                opacity: 85%;
                border-radius: 12px;
                margin-left: 25px;
                margin-bottom: -20px;" href="https://www.facebook.com" class="fa
fa   facebook"></a>
          <a style:"color:white;
                padding: 20px;
                background-color: black;
                opacity: 85%;
                border-radius: 12px;
                margin-left: 25px;
                margin-bottom: -20px;" href="https://www.twitter.com" class="fa fa-twitter"></a>
          <a style:"color:white;
                padding: 20px;
                background-color: black;
```

```
                opacity: 85%;
                border-radius: 12px;
                margin-left: 25px;
                margin-bottom: -20px;" href="https://www.linkedin.com" class="fa
fa   linkedin"></a>
          <a style:"color:white;
                padding: 20px;
                background-color: black;
                opacity: 85%;
                border-radius: 12px;
                margin-left: 25px;
                margin-bottom: -20px;" href="https://www.instagram.com" class="fa
fa   instagram"></a>
       </center>
                  </div><br><br><br>
         <div class="end">
           <p style="color:rgb(255, 246, 246);
                  margin-top: 20px;
                  text-align: center;">
           <b> Copyright &#169;  2022. All Rights Reserved</b>
            </p>
   <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>
     </html>
```

FEATURE 2

PREDICTION

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <title> PREDICTION </title>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font   awesome/4.7.0/css/font-awesome.min.css">
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    <style>
       body{
          margin: 0;
          padding: 0;
          font-family:Arial, Helvetica, sans-serif
       }
       nav{
          position:relative;
          top: 0;
          left: 0;
          width: 100%;
          height: 70px;
          padding: 10px 100px;
          box-sizing:border-box;
       }
       nav .logo{
          padding: 15px;
          height: 30px;
          float: left;
          font-size: 25px;
          font-weight: bold;
          color: #fff;
       }
       nav ul {
          list-style:none;
          float: right;
          margin: 0;
          padding: 0;
          display: flex;
          font-weight: 600;
       }
       nav ul li a{
          float: right;
          display: block;
          color: #f2f2f2;
          text-align: center;
```

```
            padding: 15px;
            text-decoration: none;
            font-size: 17px;
        }
        nav ul li a:hover{
            background: rgb(200, 212, 200);
            border-radius: 6px;
            color: rgb(51, 48, 47);
        }
        nav ul li a.active{
            background: #e2472f;
            border-radius: 6px;

        }
        body {
         margin:0;
         padding:0;
         font-family: sans-serif;
         background: linear-gradient(to top,
rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%),url("/static/images/2.jpg");
         background-position: center;
         background-size: cover;
          height: 100vh;
}
        .end {
            overflow: hidden;
            background-color: rgb(63, 63, 63);
            position: fixed;
            bottom: 0;
            height: 55px;
            width: 100%;
   }
   button{
                background-color:black;
                opacity: 78%;
                color: rgb(255, 255, 255);
                font-size: 20px;
                border-radius: 50px;
```

```
                width: 150px;"
        }
  input{
                background-color:black;
                opacity: 78%;
                color: white;
                font-size: 15px;
                width: 250px;
                  }
 </style>
  </head>
  <body style="background image=/static/images/2.jpg;">
      <div class="wrap">
      <nav>
      <div class="logo">PREDICTION  PAGE</div>
      <ul>
         <li><a href="{{ url_for('dashboard') }}">Home</a></li>
         <li><a href="{{ url_for('logout') }}">Logout </a></li>
      </ul>
      </div>
      </nav><br><br><br><br><br><br><br><br><br><br>
   <div style="margin-top: -45px;" class="container">
    <center>
      <h2 style="font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif; color:
#ff7200;"><b>UPLOAD IMAGE TO PREDICT</b></h2><br>
    <form action="{{url_for('prediction')}}" method="POST"  enctype="multipart/form-data">
      <div class="input">
                  <input type="file" class="myFile" id="myFile" name="myFile">
                  </div>
      <br><br><br>
     <div class="button">
         <button input type="submit" >Submit</button></div>
    </form>
          <br>
   <!-- <script>
     function flashMessage(){
      if("{{ flash_message }}"=='True'){
        const im = document.createElement('img');
```

```
        im.src = "{{url_for('static', filename='imagedata/save.png')}}";
        im.height = "200px";
        im.width = '200px';
        im.alt = 'hello world'
       }
     }
   </script> -->
  <!-- <img src="{{url_for('static', filename='imagedata/save.png')}}" alt=""
   height="200px"
   width="200px"
   /> --><br><h2 style="font-family:Arial, Helvetica, sans-serif; color: #ff7200;" > <b> The Estimated cost for the Damage is = {{prediction_text}}</b></h2>
        </h3>
   </center>
</div>
</body>
</html>
```

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 validatea particular test objective/target, which when followedwill tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

1. Accurate: Exacts the purpose.
2. Economical: No unnecessary steps or words.
3. Traceable: Capable of being traced to requirements.
4. Repeatable: Can be used to perform the test over and over.
5. 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

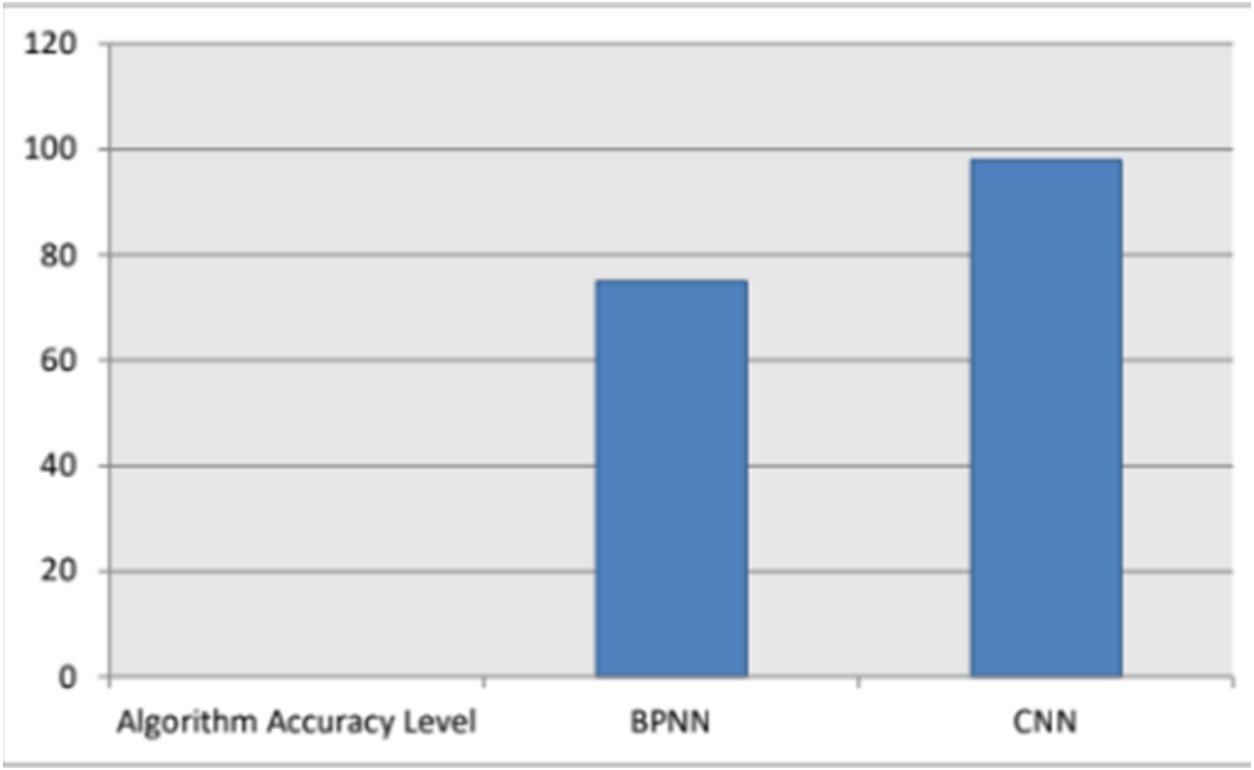
This sort of testingis carried out by users,clients, or other authorised bodiesto identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application 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

Theresult represented for the projectthat can get the desiredoutput for the particular project and it is used to check if the input and the output are valid or invalid. Data, reports, Deliverables, and any other Know-How developed or produced in the course of Development Activities performed under any Project Schedule; excluding Inventions.

PERFORMANCE METRICS

Project metrics are used to track the progress and performance of a project. Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track. As a project evolves, managers need access to changing deadlines or budgets to meet their client's expectations.



ADVANTAGES & DISADVANTAGES

ADVANTAGE

1. Digitalized claim processmakes easy to use
2. Give the accurateresult of the damaged vehicle
3. Helps the insurance companyto analyze the damaged vehicleand also payment process.

DISADVANTAGE

4. It will take more time to claimthe insurance in manual process
5. Because of incorrect claims,the company behavesbadly and doesn’tmake payments currently.
6. 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 regularization methods with a big datasetin our next work. Anticipate the cost of a car damaged componentmore 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 validatethe 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.

In the future, we will continueto explore the innovation of insurance technologyof 'AI + Vehicle Insurance'. We hope that we can use the power of intelligent damage determination system. On the one hand, the owner can take photos by one click to achieve rapid loss determination, price estimation and immediate compensation. On the other hand, it assists insurance companies to achieve rapid and accurate pricing in the process of fixing losses and claims. Finally, by combining the rapid compensation of accident vehicles to relieve traffic pressure, to avoid more serious personal and property losses caused by secondary accidents.

APPENDIX

SOURCE CODE

```
from flask import Flask, app, request, render_template

import os

import flask

import re

import flask_login

import base64

from PIL import Image

from io import BytesIO

import datetime

import cv2

import numpy as np

from tensorflow.keras.models import load_model

from cloudant.client import Cloudant

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

model1 = load_model('Model/level.h5')

model2 = load_model('Model/body.h5')

def detect(frame,model1,f):
```

```
    img = cv2.resize(frame,(244,244))
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    if(np.max(img)>1):
        img=img/255.0
```



```
    img = np.array([img])
    prediction = model1.predict(img)
    if(f):
        label= ['front','rear','side']
    else:
        label =['minor','moderate','severe']
    preds = label[np.argmax(prediction)]
    return preds
client = Cloudant.iam('862d725c-4fb3-4619-bfcb-
083c11c6a979-
bluemix','QM5pS9ePdxMpe6Lh-
8yIvNYoZ3SXtoIdQQKnyIRYlwFb',connect=True)
name = 'name'
email = 'a@b.c'
password = '123'
user_database = client.create_database('user_database')
```

```
user_image_database =
client.create_database('user_image_database')
def image_database_updation(name,email,imagestr):
    global user_image_database
    now = datetime.datetime.now()
    json_image_document={
        'name':name,
        'email':email,
        'image':imagestr,
        'datetime':now.strftime("%m/%d/%Y, %H:%M:%S")
    }
    new_image_document  =
user_image_database.create_document(json_image_docu
ment)
```



```
    if(new_image_document.exists()):
        print('database updated')
    else:
        print('database couldn\'t be edited')
    return
def image_database_retrieval():
```

```
    global user_image_database
    image_result_retrieved =
Result(user_image_database.all_docs,include_docs=True)
    image_result ={}
    for i in image_result_retrieved:
        if(i['doc']['email'] in image_result.keys()):
            # like current date> rx date('str')
            n =
datetime.datetime.strptime(i['doc']['datetime'],'%m/%d/%
Y, %H:%M:%S')
            o =
datetime.datetime.strptime(image_result[i['doc']['email']]['
date'],'%m/%d/%Y,
%H:%M:%S')
            if(n>o):
                image_result[i['doc']['email']] =
{'name':i['doc']['name'],'image':i['doc']['image'],'date':i['doc
']['datetime']}
        else:
            image_result[i['doc']['email']] =
{'name':i['doc']['name'],'image':i['doc']['image'],'date':i['doc
']['datetime']}
```

```
    return(image_result)
def database_updation(name,email,password):
    global user_database
    jsonDocument = {
        'name':name,
44
        'email':email,
        'password':password
    }
    newDocument =
user_database.create_document(jsonDocument)
    if(newDocument.exists()):
        print('database updated')
    else:
        print('database couldn\'t be edited')
    return
#database_updation(name,email,password)
def database_retrieval():
    global user_database
    result_retrieved =
Result(user_database.all_docs,include_docs=True)
```

```
    #print(list(result_retrieved))
    result = {}
    for i in list(result_retrieved):

result[i['doc']['email']]={'name':i['doc']['name'],'password':i
['doc']['password']}
    return result
#print(database_retrieval())
app = Flask(__name__)
app.secret_key = 'apple'
login_manager = flask_login.LoginManager()
login_manager.init_app(app)
users = {'a@b.c': {'password': '123'}}
```



```
class User(flask_login.UserMixin):
    pass
@login_manager.user_loader
def user_loader(email):
    data = database_retrieval()
    if email not in data:
        return
```

```
    user = User()
    user.id = email
    user.name = data[email]['name']
    return user
@login_manager.request_loader
def request_loader(request):
    email = request.form.get('email')
    data = database_retrieval()
    if email not in data:
        return
    user = User()
    user.id = email
    user.name = data[email]['name']
    return user
@app.route('/')
def index():
    if(flask_login.current_user.is_authenticated):
        return render_template('dashboard.html')
```



```
    else:
        return flask.redirect(flask.url_for('login'))
```

```
@app.route('/register',methods = ['GET','POST'])
def register():
    data = database_retrieval()
    if(flask.request.method == 'GET'):
        return render_template('register.html')
    email = flask.request.form['email']
    if(email in data):
        return
render_template('register.html',flash_message='True')
    else:

database_updation(flask.request.form['name'],email,flask.r
equest.form['password'])

#users[email]={'password':flask.request.form['password']}
        user  = User()
        user.id = email
        user.name = flask.request.form['name']
        flask_login.login_user(user)
        return
render_template('dashboard.html',flash_message='True')
@app.route('/login',methods =['GET','POST'])
```

```
def login():
    data = database_retrieval()
    if(flask.request.method == 'GET'):
            return
render_template('login.html',flash_message='False')
    email = flask.request.form['email']
    if(email in data and
flask.request.form['password']==data[email]['password']):
```



```
        user  = User()
        user.id = email
        flask_login.login_user(user)
        return
render_template('dashboard.html',flash_message='Fal')
    #flask.flash('invalid credentials !!!')
    return
render_template('login.html',flash_message="True")
    #error = 'inavlid credentials')
@app.route('/dashboard',methods = ['GET','POST'])
@flask_login.login_required
def dashboard():
    if(flask.request.method == 'GET'):
```

```
        return
render_template('dashboard.html',flash_message='False')
    email = flask.request.form['email']
    if(email in users and
flask.request.form['password']==users[email]['password']):
        user  = User()
        user.id = email
        flask_login.login_user(user)
        return
render_template('dashboard.html',flash_message="Fal")
    return
render_template('dashboard.html',flash_message="Fals")
@app.route('/logout')
@flask_login.login_required
def logout():
    flask_login.logout_user()
    return render_template('logout.html')
```



```
@app.route('/prediction',methods = ['GET','POST'])
@flask_login.login_required
def prediction():
    from tensorflow.keras.models import load_model
```

```
    model1 = load_model('Model/level.h5')
    model2 = load_model('Model/body.h5')
    if(flask.request.method=='POST'):
        img = flask.request.files['myFile']
        try:
            os.remove('static\imagedata\save.png')
        except:
            pass
        imgstr = base64.b64encode(img.read()).decode('utf-
8')

image_database_updation(flask_login.current_user.name,f
lask_login.current_user.id,imgstr)
        data = image_database_retrieval()
        print(flask_login.current_user.id)

#print(len(base64.b64decode(data[flask_login.current_use
r.id]['image'].strip())))
        image =
Image.open(BytesIO(base64.b64decode(data[flask_login.c
urrent_user.id]['image'])))
        img_retrived = np.array(image)
```

```
        '''img_retrived =
np.asarray(base64.b64decode(data[flask_login.current_use
r.id]['image']))
        print(data[flask_login.current_user.id]['image'])
        print(img_retrived.shape)'''
        #img_retrived = np.resize(img_retrived,(244,244))
49
        img_retrive = Image.fromarray(img_retrived)
        img_retrive.save('static\imagedata\save.png')
        '''img_retrived = np.frombuffer(
            BytesIO(

base64.b64decode(data[flask_login.current_user.id]['imag
e'])
                )
            )'''
        print('################################')
        result1=detect(img_retrived,model1=model2,f=True)
        result2 =
detect(img_retrived,model1=model1,f=False)
        value=''
        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 = '900 - 11000 INR'
    elif(result1 == 'side' and result2 == 'severe'):
        value = '12000 - 15000 INR'
    else:
        value = '16000 - 50000 INR'
    print(result1,result2,value)
```

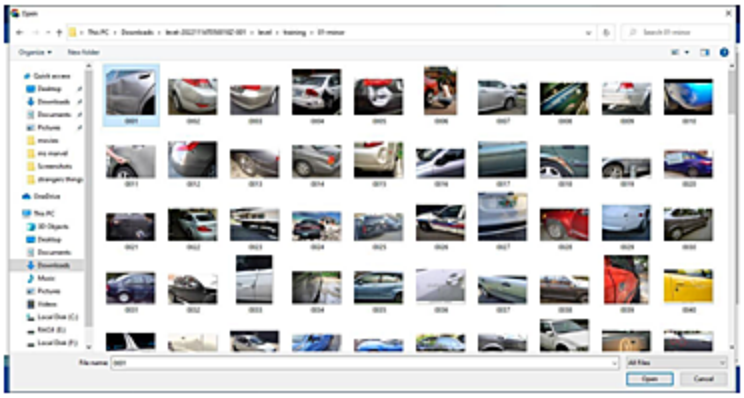
```
        print('###################################')
        img_retrived = Image.fromarray(img_retrived)
        img_retrived.save('static\imagedata\save.png')
        print('image uploaded and retrieved')
        return
render_template('prediction.html',prediction_text='{}'.for
mat(value),flash_message='False')
    return
render_template('prediction.html',flash_message='True')
if __name__ == '__main__':
    app.run(debug=True)
```

WELCOME PAGE OF WEBPAGE:

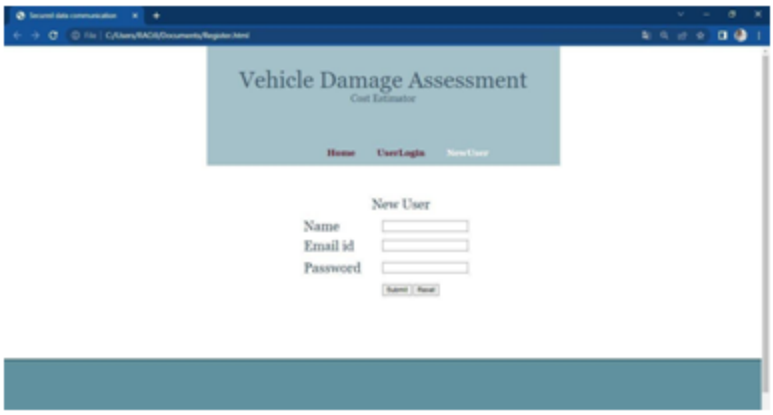
USER LOGIN PORTAL:

UPLOAD THE IMAGES:

SELECTING THE IMAGES TO UPLOAD:



NEW USER LOGIN PORTAL:



EXIT FROM THE PORTAL:

GITHUB AND PROJECT LINK

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-50984-1660962138

PROJECT DEMO LINK:

https://drive.google.com/file/d/1HOTqDwLWM10-FEdxGfbL0IhlnRzrqBYL/view?usp=share_link