INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

ABSTRACT

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims. The aim of this project is to build a VGG16 model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pics and the model can assess damage(be it dent scratch from and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

1. INTRODUCTION

1.1 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.

1.2 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 SURVEY

Auto Insurance Claim Using CNN Model

Li Ying & Dorai Chita, presented the CNN Model for the auto insurance claims process, improvements in the First Notice of Loss and rapidity in the investigation and evaluation of claims could drive significant values by reducing loss adjustment expense. This paper proposed a novel application where advanced technologies in image analysis and pattern recognition are applied to automatically identify and characterize automobile damage.
Success in this will allow some cases to proceed without human adjusters, while others to proceed more efficiently, thus ultimately shortening the time between the first Notice of Loss and the final pay-out. To investigate its feasibility. they built a prototype system which automatically identifies the damaged area(s) based on the comparison of ages. Performance of the before- and after-accident automobile in of the prototype system has been evaluated on images taken from forty scaled model cars under reasonably controlled environments, and encouraging results were obtained. It is a belief that, with the advancement of image analysis and pattern recognition technologies, their proposed idea could evolve into a very promising application area where the auto insurance industry could significantly benefit. The main drawback in this model was that the automobile damaged can be analyzed only having white background otherwise it will be not able to give the desired results and the study also indicates that there may be an error in the result, it may not give that accurate result like 85-90% affective.

Image Based Vehicle insurance

U. Waqas, N. Akram, S. Kim, D. Lee and J. Jeon, they presented the Image- based vehicle insurance processing and loan management has large scope for automation in automotive industry. In this paper consideration of the problem of car damage classification, where categories include medium damage, huge damage and no damage. Based on deep learning techniques, Mobile Net model is proposed with transfer learning for classification. Moreover, moving towards automation also comes with diverse hurdles; users can upload fake images like screenshots or taking pictures from computer screens, etc. To tackle this problem a hybrid approach is proposed to provide only authentic images to algorithm for damage classification as input. In this regard, moiré effect detection and metadata analysis are performed to detect fraudulent images. For damage classification 95% and for moiré effect
detection 99% accuracy is achieved. The main drawback was that Images in bad lighting, awkward angles, variety in vehicle models, images taken in rain or snow, minor scratches on vehicles, etc. Even though it used several angles and vehicle models in a small dataset to achieve automation but still the range is broad.

Damage Analysis of AI based Machine Learning

Phyu Mar Kyu and Kuntpong Woraratpanya they presented the sense of Artificial Intelligence (AI) based on machine learning and deep learning algorithms which can help to solve the problem for insurance industries for damage analysis. In this paper, they applied deep learning-based algorithms, VGG16 and VGG19, for car damage detection and assessment in real world datasets. The algorithms detect the damaged part of a car and assess its location and then its severity. Initially, it discovers the effect of domain-specific pre-trained CNN models, which are trained on an ImageNet dataset, and followed by fine-tuning, because some of the categories can be fine granular to get a specific task. Then it applies transfer learning in pre-trained VGG models and use some techniques to improve the accuracy of the system. To achieve the accuracy of 95.22% of VGG19 and 94.56% of VGG16 in the damaged detection, the accuracy of 76.48% of VGG19 and 74.39% of VGG16 in damage localization, the accuracy of 58.48% of VGG19 and 54.8% of VGG16 in damage severity with the combination of transfer learning and L2 regularization. From their results, the performance of VGG19 is better than VGG16. After analysing and implementing the models, it finds out that the results of using transfer learning and L2 regularization can work better than those of fine-tuning. The drawback of this model was since car damaged assessment is a specific domain, it is lack of publicly available datasets for car damaged images with labelling. Training a model with a small dataset is the most challenging.

Damage Detection at Deep learning based Architecture

Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud they presented automated and efficient deep learning-based architectures for vehicle: damage detection and localization. The proposed solution combines deep learning, instance segmentation, and transfer learning techniques for features extraction and damage identification. Its objective is to automatically detect damages in vehicles, locate them, classify their severity levels, and visualize them by contouring their exact locations. Numerical results reveal that our transfer learning proposed solution, based on Inception-Resnet V2 pre- trained model followed by a fully connected neural network, achieves higher performances in features extraction and damage detection/localization than another pre trained model, i.c.. VGG16. The transfer learning could significantly reduce the training times when it uses the weights of pre trained VGG models. Furthermore, it had demonstrated significant progress on how to solve classification problems when the small dataset was not enough to train a CNN model. The classes of the pre-trained VGG models are the source tasks, and the detected damaged parts of their locations, and their damaged levels are the target tasks in our system. The main drawback of this model was A reduction of model training time is also the most challenge. Typically, a traditional CNN model can be very time- consuming to perform image classification tasks and identify the correct weights for the network by multiple forward and backward iterations. This process may take days or even weeks to complete it using GPUs.

Mask R-CNN

Mask RCNN is a deep neural network aimed to solve instance segmentation problem in machine learning or computer vision. In other words, it can separate different objects in an image or a video. You give it an image, it gives you the object bounding boxes, classes and masks.There are two stages of Mask RCNN. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object, refines the bounding box and generates a mask in pixel level of the object based on the first stage proposal. Both stages are connected to the backbone structure [1].

Car Damage Assessment using CNN

The project involved developing and training a CNN model with 10 convolutional layers and 3 pooling layers with Relu as the activation function at each layer and the final layer being a Fully Connected Layer. The dataset used in training the model was obtained through web scraping. The model performed well on high quality images hut gave inaccurate results on blurred images. The main disadvantage was the back of widely available labelled dataset [2].

Automatic Car Damage Assessment through videos

Wei Zhang, Yuan Cheng, Xin Guo, Qingpei Guo, Jian Wang, Qing Wang, Chen Jiang, Meng Wang, Furong Xu, Wei Chu proposed a method to detect and analyze car damage through user input videos. The approach involved 2 modules Damage recognition and localisation and component recognition and localisation to segment the damage and components at pixel level to get accurate results. The model required high quality videos as input to generate accurate results [3].

Recognition of Car Manufacturers using Faster R-CNN and Perspective Transformation

Israfil Ansari, Yeunghak Lee, Yunju Jeong, Jaechang Shim proposed a method to detect car logos from CCTV footages. The approach involved performing perspective transformation on CCTV footages to get a clear view of the logos and then detecting and localizing the car logos through faster RCNN [4].

Vehicle Logo Detection and Classification using Discriminative Pixel- patches Sparse Coding

Yi Ouyang developed a system to detect and classify vehicle logos with the help of sparse coding. The method localised the car logos by detecting the number plate with the help of 3-channel pixel regression technique then performing multi class structural linear SVM for logo classification [5].

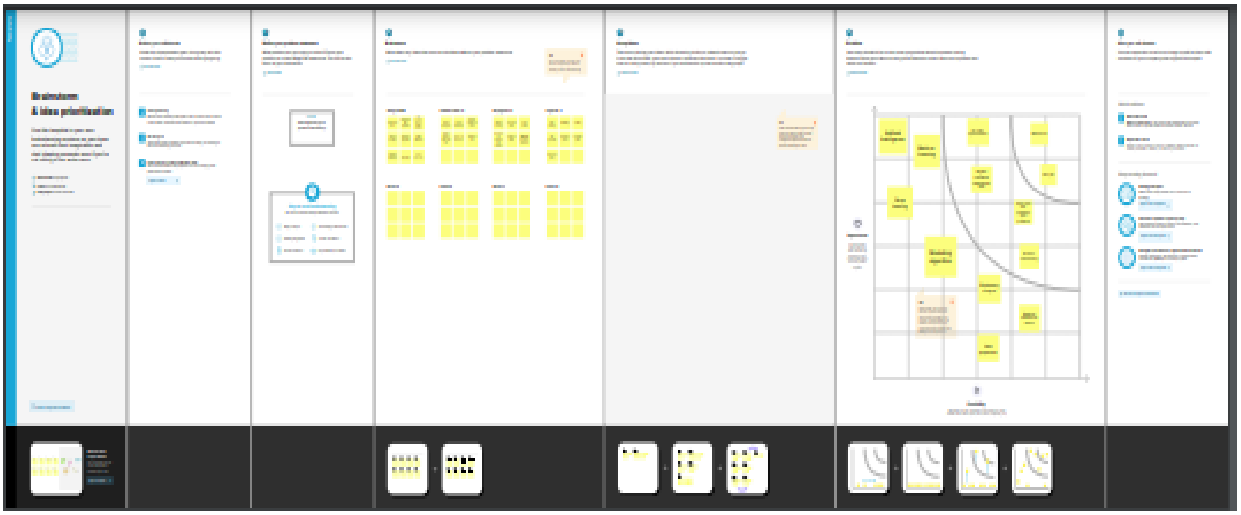
Vehicle Type classification With Deep Learning

The paper researches various algorithms to classify the car body type from images as SUV, sedan, pick-up truck. Dataset used was the stanford dataset with 224 images and achieved an accuracy of 76 percent when arithmetic mean computation was on a hierarchical tree on ResNet 34 architecture [6].

2.3 PROBLEM STATEMENT DEFINITION

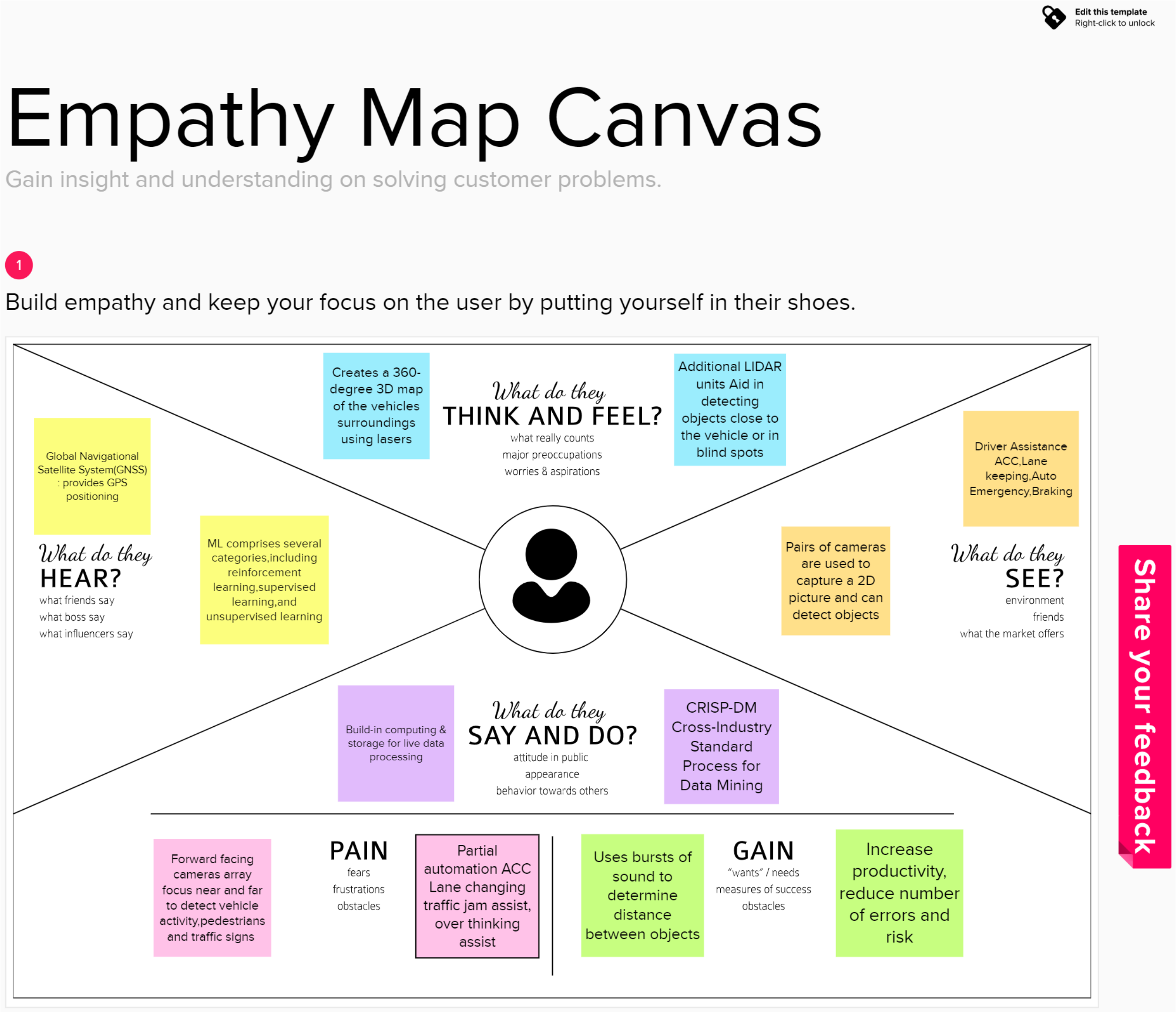
In existing system, the procedure of making an insurance claim for an automobile is laborious, and there is a delay before the first reimbursement is authorised. Insurance firms lose millions of dollars each year due to claim leakage as a result of the expansion of the vehicle sector and the daily rise 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.

3. IDEATION & PROPOSED SOLUTION



Brainstorm
& idea prioritization

EMPATHY MAP CANVAS



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

Creates a 360-degree 3D map of the vehicles surroundings using lasers

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Additional LIDAR units Aid in detecting objects close to the vehicle or in blind spots

Global Navigational Satellite System(GNSS) : provides GPS positioning

What do they
HEAR?
what friends say
what boss say
what influencers say

ML comprises several categories,including reinforcement learning,supervised learning,and unsupervised learning

Driver Assistance ACC,Lane keeping,Auto Emergency,Braking

Pairs of cameras are used to capture a 2D picture and can detect objects

What do they
SEE?
environment
friends
what the market offers

Build-in computing & storage for live data processing

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

CRISP-DM Cross-Industry Standard Process for Data Mining

Forward facing cameras array focus near and far to detect vehicle activity,pedestrians and traffic signs

PAIN
fears
frustrations
obstacles

Partial automation ACC Lane changing traffic jam assist, over thinking assist

Uses bursts of sound to determine distance between objects

GAIN
"wants" / needs
measures of success
obstacles

Increase productivity, reduce number of errors and risk

3.3 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.

3.4 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%.

4. REQUIREMENT ANALYSIS

4.1 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'sdamaged 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 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. 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%.

4.2 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.

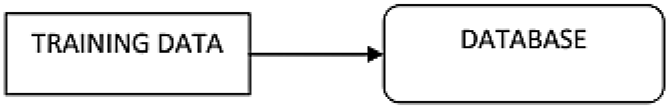
5. PROJECT DESIGN

5.1 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.

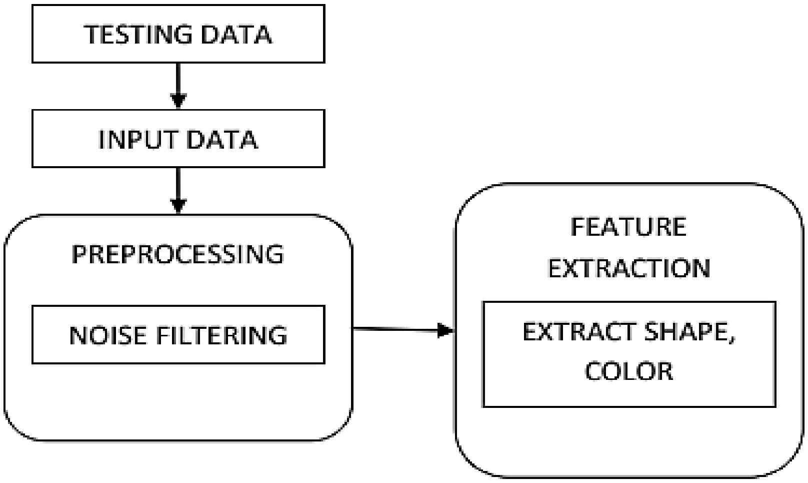
LEVEL 0

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.



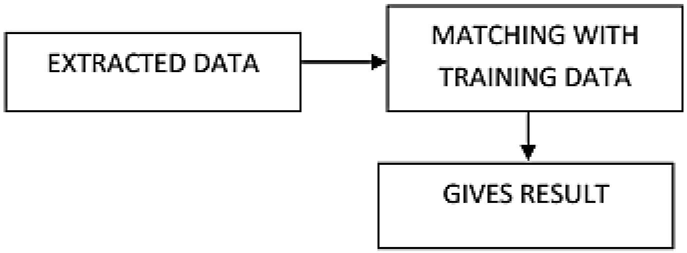
LEVEL 1

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.



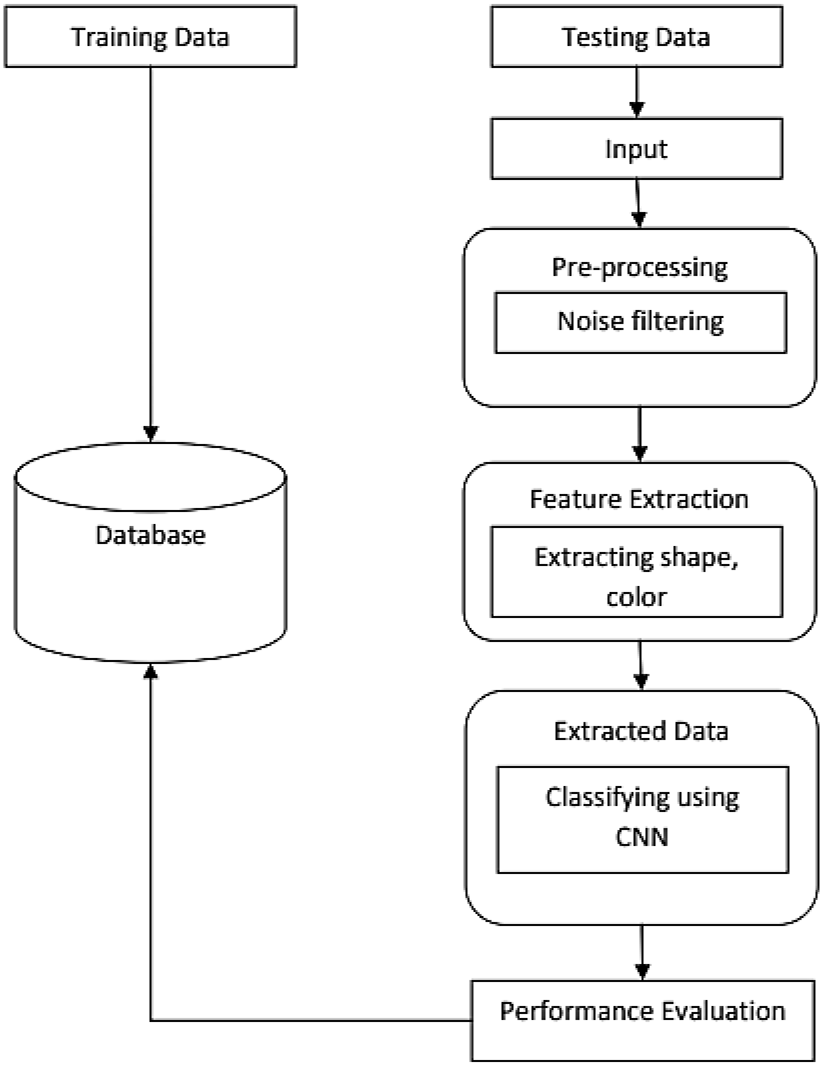
LEVEL 2

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.



5.2 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).


Training Data
Testing Data
Input
Pre-processing
Noise filtering
Database
Feature Extraction
Extracting shape, color
Extracted Data
Classifying using CNN
Performance Evaluation

5.3 USER STORIES

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

6.2 SPRINT DELIVERY SCHEDULE

6.3 REPORTS FROM JIRA

7. CODING & SOLUTIONING

7.1 FEATURE 1

7.2 FEATURE 2

7.3 DATABASE SCHEMA

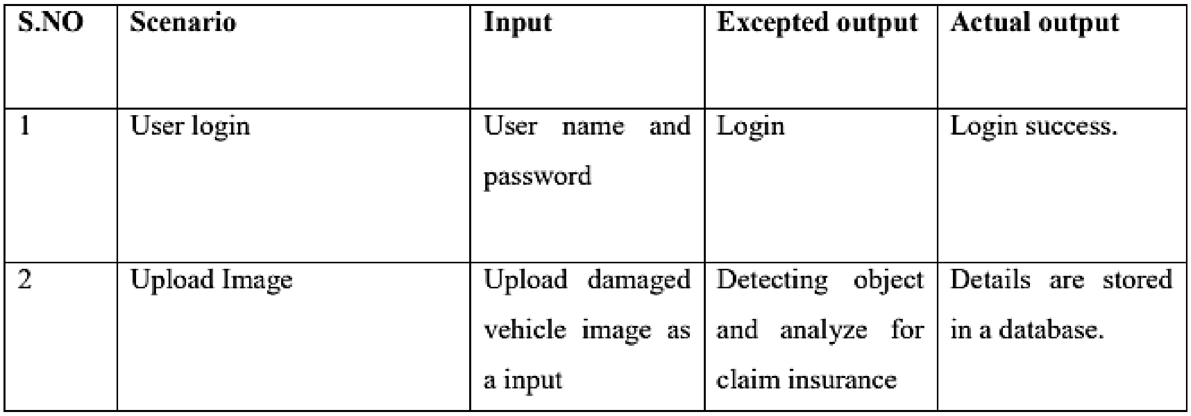
8. TESTING

8.1 TEST CASES

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

Characteristics of a good test case:

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



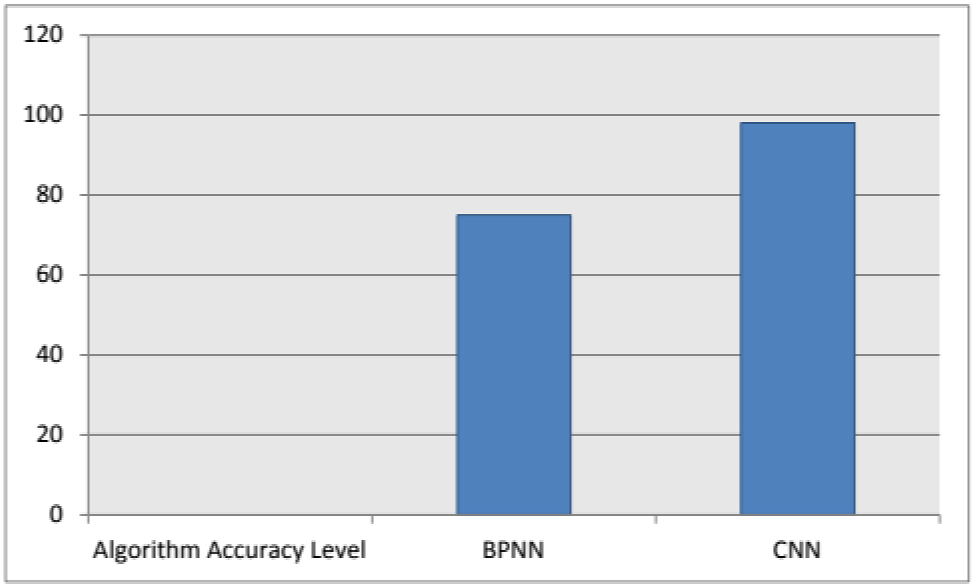
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload damaged vehicle image as a input	Detecting object and analyze for claim insurance	Details are stored in a database.

8.2 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).

9. RESULTS

9.1 PERFORMANCE METRICS



10. 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

11. CONCLUSION

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

12. FUTURE SCOPE

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

13. APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request,session
from cloudant.client import  Cloudant
import cv2
client = Cloudant.iam("eb55a2b7-ae45-4df8-8d1c-69c5229ffdbe-
bluemix","YzG5FZg9Vs_HScOBZaWyVXm7PpNjbPrmPaPMfHx7w3X9",connect=
True)
my_database = client.create_database("database-dharan")
app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/userhome")
def userhome():
    return render_template('userhome.html')
@app.route("/addamount")
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')
@app.route("/user")
def user():
    return render_template('user.html')
@app.route("/newuse",methods=['GET','POST'])
def newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {'_id': x[1],
        'name': x[0],
        'psw': x[2]
        }
    print(data)
    query = {'_id': {'Seq': data['_id']}}
    docs = my_database.get_query_result(query)
    print(docs)
    print(len(docs.all()))
    if (len(docs.all()) == 0):
        url = my_database.create_document(data)
        return render_template('goback.html', data="Register, please login using your
details")
    else:
        return render_template('goback.html', data="You are already a member, please
login using your details")
@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':
    user = request.form['_id']
    passw = request.form['psw']
    print(user, passw)
    query = {'_id': {'$eq': user}}
    docs = my_database.get_query_result(query)
    print(docs)
    print(len(docs.all()))
    if (len(docs.all()) == 0):
        return render_template('goback.html', pred="The username is not found.")
    else:
        if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):
            return render_template("userhome.html")
        else:
            return render_template('goback.html',data="user name and password
incorrect")
@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        file.save('static/Out/Test.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test.jpg')
```

```
# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)
result1 = "
if result[0][0] == 1:
        result1 = "front"
elif result[0][1] == 1:
        result1 = "rear"
elif result[0][2] == 1:
        result1 = "side"
file = request.files['fileupload1']
file.save('static/Out/Test1.jpg')
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
classifierLoad = tf.keras.models.load_model('level.h5')
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('static/Out/Test1.jpg', target_size=(200, 200))
img1 = cv2.imread('static/Out/Test1.jpg')
#test_image= image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)
result2 = "
if result[0][0] == 1:
        result2 = "minor"
elif result[0][1] == 1:
        result2 = "moderate"
elif result[0][2] == 1:
        result2 = "severe"
if (result1 == "front" and result2 == "minor"):
        value = "3000 - 5000 INR"
elif (result1 == "front" and result2 == "moderate"):
        value = "6000 8000 INR"
elif (result1 == "front" and result2 == "severe"):
        value = "9000 11000 INR"
elif (result1 == "rear" and result2 == "minor"):
        value = "4000 - 6000 INR"
elif (result1 == "rear" and result2 == "moderate"):
        value = "7000 9000 INR"
elif (result1 == "rear" and result2 == "severe"):
```

```
            value = "11000 - 13000 INR"
        elif (result1 == "side" and result2 == "minor"):
            value = "6000 - 8000 INR"
        elif (result1 == "side" and result2 == "moderate"):
            value = "9000 - 11000 INR"
        elif (result1 == "side" and result2 == "severe"):
            value = "12000 - 15000 INR"
        else:
            value = "16000 - 50000 INR"
        return render_template('userhome.html', prediction=value)
    if __name__ == '__main__':
app.run(debug=True, use_reloader=True)
```

GITHUB & PROJECT DEMO LI