



Car Damage Assessment & Cost Estimator



IBM NALAIYA THIRAN

PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID30946

| | |
|------------------------|-----------------------|
| ABHAY KUMAR | (620819104001) |
| SAURAV KUMAR | (620819104097) |
| MD SAJID IQUBAL | (620819104059) |
| SAJAN KUMAR | (620819104086) |
| OM KUMAR TIWARI | (620819104072) |

INDUSTRY MENTOR

(SWATHI)

FACULTY MENTOR

(R.UMAMAHESHWARI)



Table of Contents

| | |
|---------------------------------------------------|-----------|
| 1. INTRODUCTION | 3 |
| 1.1 Project Overview | 3 |
| 1.2 Purpose | 3 |
| 2. LITERATURE SURVEY..... | 4 |
| 2.1 Existing Problem..... | 4 |
| 2.2 References | 4 |
| 2.3 Problem Statement Definition | 8 |
| 3. IDEATION & PROPOSED SOLUTION..... | 9 |
| 3.1 Empathy map canvas | 9 |
| 3.2 Ideation & Brainstorm | 10 |
| 3.3 Proposed Solution | 13 |
| 3.4 Problem Solution fit..... | 14 |
| 4. REQUIREMENT ANALYSIS..... | 15 |
| 4.1 Functional Requirement | 15 |
| 4.2 Non-Functional Requirements..... | 15 |
| 5. PROJECT DESIGN | 16 |
| 5.1 Data Flow Diagrams | 16 |
| 5.2 Solution & Technical Architecture..... | 16 |
| 5.3 User Stories | 17 |
| 6. PROJECT PLANNING & SCHEDULING | 18 |
| 6.1 Sprint Planning & Estimation..... | 18 |
| 6.2 Sprint Delivery Schedule | 20 |
| 7. CODING AND SOLUTIONING | 21 |
| 7.1 Feature 1 | 21 |
| 8. TESTING..... | 23 |
| 8.1 Test Cases..... | 23 |
| 8.2 User Acceptance Testing..... | 30 |
| 9. RESULTS | 31 |
| 9.1 Performance Metrics | 31 |
| 10. ADVANTAGES AND DISADVANTAGES..... | 33 |
| 11. CONCLUSION..... | 34 |
| 12. FUTURE SCOPE | 35 |
| 13. APPENDIX..... | 36 |

Source Code

GitHub & Project link Demo

1. INTRODUCTION

1.1 Project Overview

Analysis of the damaged vehicle that can be automatically claiming insurance that takes human resource, time and effort. Image processing and machine learning techniques are analysing the vehicle damage in the proposed solution. In Advanced solution helps to speed up the claiming process sufficiently. Consider a situation, if a person is driving a car they met an accident the vehicle owner can take a few photos of the damaged car from a mobile phone that can be send to the insurance company and can just upload the photos to the system. The system can analyse the damage, severity of the damage as well as location of the damage. In this proposed project the insurance company can machine-driven the car damage analysis process without the need for humans to analyse the damage done to the car. Therefore, it is a very challenging task for quality of computer vision techniques and also Machine learning technologies

1.2 Purpose

By the end of the project we can:

- ✓ Know fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks.
- ✓ Know fundamental concepts and techniques of the python for machine learning.
- ✓ Gain a broad understanding of image data.
- ✓ Work with Sequential type of modelling.
- ✓ Work with Keras capabilities.
- ✓ Work with image processing techniques.
- ✓ Know how to build a web application using the Flask framework.

2. LITERATURE SURVEY

2.1 Existing Problem

- No FIR
- Hand Lack of proper documentation
- Delayed Application.
- Inadequate insurance coverage
- Insufficient insurance knowledge
- Text in handwriting can have variable rotation to the right which contrasts with printed text where all the text sits up straight.
- Collecting a good, labelled dataset to learn is not cheap compared to synthetic data.

2.2 References

- 1. A.Neela Madheswari, J.haripriya, G.Kiruthika, R.M.Meyammai Mahendra Engineering college, India,exterior vehicular damage detection using deep learning, department of computer science and engineering.**
- 2. Girish N, Mohammed Aqeel Arshad, car damage detection using machine learning. International journal of advances research in computer and communication engineering, Vol. 10, issue 8, August 2021 DOI 10.17148/IJARCCE.2021.10808.**
- 3. Phyu Mar Kyu ,car damage detection and classification, faculty of information technology king Mongkut's institute of technology ladkrabang Bangkok, Thailand 62606003@kmitl.ac.in.**
- 4. S. Jayawardena, Image based automatic vehicle damage detection. PhD thesis, College of Engineering and Computer Science (CECS), 12 2013.**
- 5. S. Gontscharov, H Baumgartel, A.Kneifel, and K.-L. Krieger, Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing," Procedia Technology, vol. 15, pp. 586 {594, 2014. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering.**

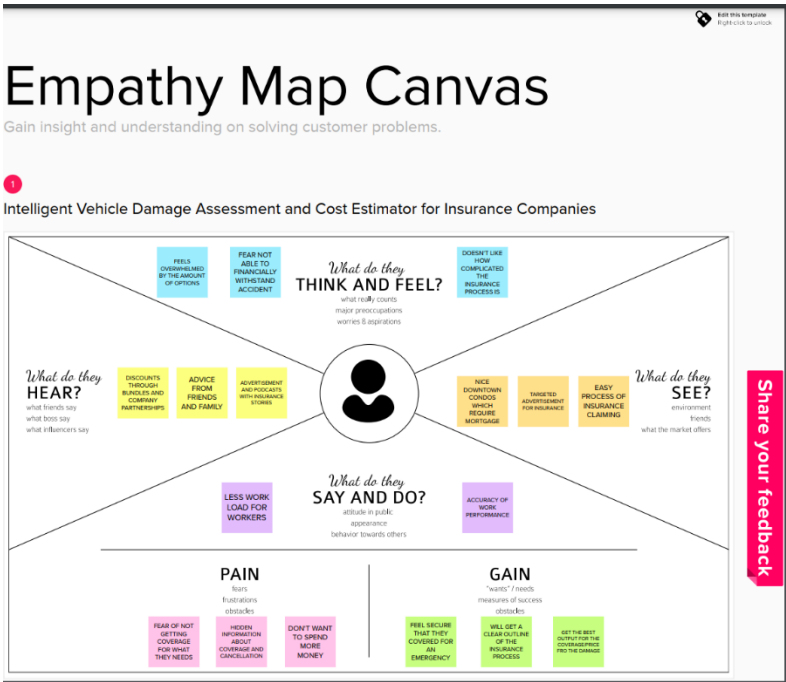
6. Y.-J. Cha, J. Chen, and O. B'uy'uk'ozt'urk, Output-only computer vision based damage detection using phase-based optical flow and unscented kalman filters," Engineering Structures, vol. 132, pp. 300, {313, 2017

2.3 Problem Statement Definition

| | |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Who does the problem affect? | Persons was claiming the assessment amount. |
| What are the boundaries of the problem? | People who vehicle and facing issues of Insurance claiming |
| What is the issue? | The vehicle is damaged, Then the next step is that the user has already insured the car with the insurance company, the customer compares the calculated amount, and the getting a lower amount, so the valuation process be seen. |
| When does the issue occur? | The issues occurred in the damage part not fully estimation interior part not estimated so issues occur in company side. |
| Where does the issue occur? | The issue occurs in Automobile industry interior part damage not fully estimated so issues occur in company side, particularly City side. |
| Why is it important that we fix the problem? | The required for Automobile industry day by developing so the opposite side accident car owner has claim the actual amount of damage so they consider to fix problem and also customer can see the each and every part repairing cost that problem also ratified in this process. |
| What solution to solve this issue? | An automated system is introduced to identifydifferent diseases on plants by checking the symptom's shown on the leaves of the plant. |
| What methodology used to solve the issue? | Deep learning techniques are used to identify the specific part of damage repair cost and suggest modifying cost rate. |

3. IDEATION & PROPOSSED SOLUTION


3.1 Empathy map canvas



3.2 Ideation & Brainstorm




Step-1: Team Gathering, Collaboration and Select the Problem Statement

Template



Brainstorm & idea prioritization


Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 10 minutes to prepare
 1 hour to collaborate
 2-8 people recommended

[Share template feedback](#)

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

 10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or prework ahead.


B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →






1 Define your problem statement


What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

 5 minutes

How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

-  Stay in topic.
-  Encourage wild ideas.
-  Defers judgment.
-  Listen to others.
-  Go for volume.
-  If possible, be visual.



Need some inspiration?

Get a head start on your work by using the template for **HOWMIGHTWE** your work.

[Open example](#) →

Step-2: Brainstorm, Idea Listing and Grouping

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

💡

You can select a sticky note and hit the pencil button to modify text to start drawing!

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

💡

Add customizable tags to sticky notes to make it easier to find, format, organize, and categorize important ideas as themes surface your ideas!

Step-3: Idea Prioritization



3.3 Proposed Solution

Project Design Phase-I Proposed Solution Template

| | |
|---------------|-----------------------------------------------------------------------------------|
| Date | 19 September 2022 |
| Team ID | PNT2022TMID30946 |
| Project Name | INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATION FOR INSURANCE COMPANIES |
| Maximum Marks | 2 Marks |

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

| S.No. | Parameter | Description |
|-------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Problem Statement (Problem to be solved) | Nowadays 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. There is no easy way of accessing and knowing about the part of the vehicle getting damaged. Often the processing of such a damaged part of the vehicle carrying the area of damaged part is cumbersome. New methods have to be proposed in order to make it faster and efficient. Processing of Insurance for the cars needs to be assessed in a quicker way so that claims can be provided to the damaged parts. |
| 2. | Idea / Solution description | Automobile Industry is one of the major industry in a Country. This proposed system is Intelligent vehicle damage assessment and cost estimator for insurance companies using computer vision in artificial intelligence. The model will predict the location of the damage as in front, side or rear, and the severity of such a damage as in minor, moderate or severe and estimate the cost of damage of both car and bike. |
| 3. | Novelty / Uniqueness | Deep learning method used to fixed the problem and then Working on with VGG16 pre-trained model by adding extra layers to increase the accuracy while implementing the project. |

| | | |
|----|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4. | Social Impact / Customer Satisfaction | <p>The model developed will be used to fix the damage caused to the Vehicle quickly so that the vehicle can be modified to the old look and also for faster processing of cost of the damage to claim insurance quickly.</p> <p>This project can be used to save time for calculating the area and level of the damage quickly such that the insurance claim can be made efficiently.</p> |
| 5. | Business Model (Revenue Model) | <p>This can also be used to help car companies as well.</p> <ul style="list-style-type: none"> • Can collaborate with insurance companies. • Can collaborate with car companies. |
| 6. | Scalability of the Solution | <p>AI guided Application provides 24/7 service to clear all customer queries and guide them through all the processes. In future, it can be scaled as per the requirements of the insurance or car company to include answers to queries related to the cost based on the inputs provided.</p> |

3.4 Problem Solution fit

Project Title: Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies

Project design Phase - 1

Team ID: PNT2022TMID30946

| | | | | |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Define CS, fill into CC | 1. CUSTOMER SEGMENT CS <ul style="list-style-type: none">Common peopleCar companiesInsurance companiesVehicle owner | 6. CUSTOMER CC <p>Trust Problem : The insurance companies Give the fake information about the estimation cost modify damage part of ensuring profit of company from user</p> <p>Anxiety : Customer concern is the level of severity of the damaged parts and fixing the extent of the damaged parts is also the customer concern.</p> | 5. AVAILABLE SOLUTIONS AS <p>By gathering information about the damage assessment from customer side and vehicle companies to understand the level of damage and cost needed to same</p> <p>Searching the through the internet such as online website get the insight about the damage</p> | Explore AS, differentiate |
| | 2. JOBS-TO-BE-DONE /a PROBLEMS J&P <ul style="list-style-type: none">Analyze the damage part and giving good cost estimation for customer satisfy and support systemProvide the good prediction system to predict the level of damage and modify the damage parts | 9. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none">Unaware about the level of damageProper information about the estimation costNot proper maintenance and accident Something etc..... | 7. BEHAVIOUR BE <p>The user does not know or is unaware how to deal with the damage part of vehicle</p> <p>Inefficient to predict the level of damage and analysis to estimate the cost the damaged part this is The behavior</p> | |
| Focus on J&P, fill into BE, understand RC | 3. TRIGGERS TR <ul style="list-style-type: none">Seeking help to identify the level of the damaged part.To help people with good prediction and analysis of level and cost for the damaged part of the vehicle. | 10. YOUR SOLUTION SL <ul style="list-style-type: none">Collecting damaged parts of vehicles and preprocessing them to identify the severity and the location of damage.Users will be able to identify the level of damage by the prediction system.Using VGG16 Model to train the damaged parts and trying to estimate the cost for the same. | 8. CHANNELS of BEHAVIOR CH <p>ONLINE</p> <ul style="list-style-type: none">Online WebsitesSocial Media Platforms <p>OFFLINE</p> <ul style="list-style-type: none">Customer throw WordsAnxiety and Unawareness | Identify strong TR & EM |
| | 4. EMOTIONS: BEFORE / AFTER EM <ul style="list-style-type: none">Before: Lack of proper knowledge ,Improper maintenance,Unawareness about the level of damage and difficulty in estimating the cost for damages.After: Good knowledge about the Vehicle parts,Efficient prediction system giving the level of damage and analyzing cost for the same. | | | |

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

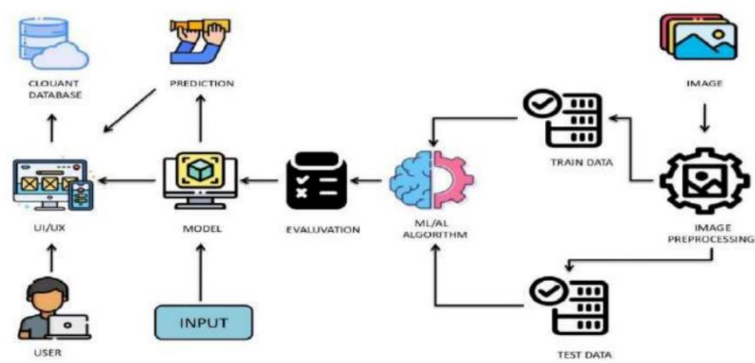
| Functional Requirement No | Functional Requirement Description |
|---------------------------|---------------------------------------------------|
| FR-1 | User Should be able to upload the image |
| FR-2 | User should be able to preview the uploaded image |
| FR-3 | Predicted number page |

4.2 Non-Functional Requirements

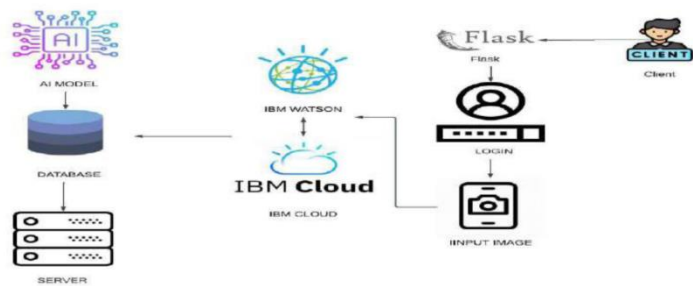
| Non-Functional Requirement No | Non-Functional Requirement Description |
|-------------------------------|----------------------------------------------------------|
| NFR-1 | The webpage should be load within 1.5 seconds |
| NFR-2 | The uploaded data should be showing the name of the file |
| NFR-3 | The website should be compatible independent of platform |
| NFR-4 | There are 500+ users can able to access the webpage |

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|------------------------|-------------------------------|-------------------|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|----------|----------|
| Customer (Mobile user) | Home page | USN-1 | As a user, I can view user guidelines and it's functionality. | I can view the guidelines | Low | Sprint-1 |
| | | USN-2 | As a user, I can read user manual to understand the process and workflow | I can read the user manual | Low | Sprint-2 |
| | | USN-3 | As a user, I can watch a video that describe about whole interface work | I can watch the video to understand the process of how to use the platform | Low | Sprint-1 |
| | Input | USN-4 | As a user, I can write the digits for prediction. | I can write digit that need to be predicted. | High | Sprint-1 |
| | Recognition | USN-6 | As a user, I will able to get the exact and accurate output | I can choose handwritten image from system and predict the output | High | Sprint-2 |
| | Predict | USN-7 | As a user, I am allowed to upload handwritten image to predict output. | I can choose the image from their own system | Medium | Sprint-3 |
| | | USN-8 | As a user, I will get the output with the help of MNIST data pre-processing to ensure maximum accurate result. | MNIST dataset provides input of handwritten digit to provide more insights. | High | Sprint-4 |
| | | USN-9 | As a user, I can view the accuracy rate of the digit predicted. | I can view accuracy rate | Medium | Sprint-3 |
| Customer (Webuser) | Access | USN-10 | As a user, I can get to use software virtually and it is user friendly. | I can view awareness of this application and its accessibility. | Low | Sprint-1 |

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|----------|-------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------|--------------|----------|----------------------------------------------------------------------------------------|
| Sprint-1 | Data Collection | USN-1 | As a user, I can collect the dataset from various resources with different handwritings. | 10 | Low | Abhay kumar Saurav kumar Md Sajid Iqubal Om kumar Tiwari Sajan kumar |
| Sprint-1 | Data Preprocessing | USN-2 | As a user, I can load the dataset, handling the missing data, scaling and split data into train and test. | 10 | Medium | Abhay kumar Saurav kumar Md Sajid Iqubal Om kumar Tiwari Sajan kumar |
| Sprint-2 | Model Building | USN-3 | As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit. | 5 | High | Abhay kumar Saurav kumar Md Sajid Iqubal Om kumar Tiwari Sajan kumar |
| Sprint-2 | Add CNN layers | USN-4 | Creating the model and adding the input, hidden, and output layers to it. | 5 | High | Abhay kumar Saurav kumar Md Sajid Iqubal Om kumar Tiwari Sajan kumar |
| Sprint-2 | Compiling the model | USN-5 | With both the training data defined and model defined, it's time to configure the learning process. | 2 | Medium | Abhay kumar Md Sajid Iqubal |
| Sprint-2 | Train & test the model | USN-6 | As a user, let us train our model with our image dataset. | 6 | Medium | Abhay kumar Md Sajid Iqubal |
| Sprint-2 | Save the model | USN-7 | As a user, the model is saved & integrated with an android application or web application in order to predict something. | 2 | Low | Abhay kumar Md Sajid Iqubal |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---------------|--------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------|---------------------|-----------------|---------------------------------------------------|
| Sprint-3 | Building UI Application | USN-8 | As a user, I will upload the handwritten digit image to the application by clicking a upload button. | 5 | High | Saurav kumar Om kumar Tiwari Sajan kumar |
| Sprint-3 | | USN-9 | As a user, I can know the details of the fundamental usage of the application. | 5 | Low | Saurav kumar Om kumar Tiwari Sajan kumar |

6.2 Sprint Delivery Schedule

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

7. CODING AND SOLUTIONING

7.1 Feature 1

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
```

```
UPLOAD_FOLDER = 'C:\\Users\\nandh\\IBM_NK\\Final Deliverables\\Final Code\\A-novel-method-for-digit-recognition-system\\flask_app\\uploads'

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("mnistCNN.h5")

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

```
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L") # convert image to monochrome
        img = img.resize((28, 28)) # resizing of input image

        im2arr = np.array(img) # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1) # printing our Labels

        return render_template('predict.html', num=str(num[0]))

if __name__ == '__main__':
    app.run(debug=False, threaded=False)
```

8. TESTING

8.1 Test Cases

| Test case ID | Feature Type | Component | Test Scenario |
|-----------------------|--------------|-------------|--------------------------------------------------------------------------|
| Homepage_TC_OO1 | Functional | Home Page | Verify user is able to see the Homepage when user enter into the website |
| Homepage_TC_OO2 | UI | Home Page | Verify the UI elements in Homepage |
| Homepage_TC_OO3 | Functional | Home page | Verify user is able to upload rather than a image file |
| Homepage_TC_OO4 | Functional | Home Page | Verify user is able to upload a image files |
| PredictionPage_TC_OO1 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO2 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO3 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |

| | | | |
|------------------------|------------|-------------|----------------------------------------------------------------------------------------------|
| PredictionPage_TC_OO4 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO5 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO6 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO7 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO8 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO9 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture |
| PredictionPage_TC_OO10 | Functional | PredictPage | Verify that the Page Shows the exact number as uploaded picture while uploading a blur image |

| Steps To Execute | Test Data | Expected Result | Actual Result | Status |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------|
| 1.Enter URL and click go 2.Verify that the routes to the specific webpage which shows the contents | https://github.com/IBM-EPBL/IBM-Project-47843-1660802851/tree/main/Final%20Deliverables/Final%20Code/A-novel-method-for-digit-recognition-system/flask_app/app.py | Homepage should display | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify Homepage with below UI elements : a.Contents b.upload button with Choose file c.A small area which showing the uploaded image d.Predict button | https://github.com/IBM-EPBL/IBM-Project-47843-1660802851/tree/main/Final%20Deliverables/Final%20Code/A-novel-method-for-digit-recognition-system/flask_app/app.py | Application should show below UI elements : a.Contents b.upload button with Choose file c.A small area which showing the uploaded image d.Predict button | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose | NK.pdf | It should Predict the Number | Working as not expected | Fail |

| | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------------------|--------------------------------|------|
| file button 3.upload a .pdf file 4.Press the predict button | | | | |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 1.png file 4.Press the predict button | 1.png | It should Predict the Number | Workin g as expecte d | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 1.png file 4.Press the predict button | 1.png | It should Predict the Number as 1 | Workin g as expecte d | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 2.png file 4.Press the | 2.png | It should Predict the Number as 2 | Workin g as expecte d | Pass |

| | | | | |
|------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------|---------------------|------|
| predict button | | | | |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 3.png file 4.Press the predict button | 3.png | It should Predict the Number as 3 | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 4.png file 4.Press the predict button | 4.png | It should Predict the Number as 4 | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 5.png file 4.Press the predict button | 5.png | It should Predict the Number as 5 | Working as expected | Pass |

| | | | | |
|------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------|---------------------|------|
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 6.png file 4.Press the predict button | 6.png | It should Predict the Number as 6 | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 7.png file 4.Press the predict button | 7.png | It should Predict the Number as 7 | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a 8.png file 4.Press the predict button | 8.png | It should Predict the Number as 8 | Working as expected | Pass |
| 1.Enter URL and click go 2.Click on Choose | 9.png | It should Predict the Number as 9 | Working as expected | Pass |

| | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------------------------------------------|-----------------------------------------|------|
| file button 3.upload a 9.png file 4.Press the predict button | | | | |
| 1.Enter URL and click go 2.Click on Choose file button 3.upload a blur.png file 4.Press the predict button | blur.png | It should Predict the Number as 9 | It returns the Numbe r as 1 | Fail |

8.2 User Acceptance Testing

DEFECT ANALYSIS

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Total |
|----------------|------------|------------|------------|------------|-------|
| By Design | 1 | 0 | 1 | 0 | 2 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 0 | 0 | 2 | 0 | 2 |
| Fixed | 4 | 1 | 0 | 1 | 6 |
| Not Reproduced | 0 | 0 | 0 | 1 | 1 |
| Skipped | 0 | 0 | 0 | 1 | 1 |
| Won't Fix | 1 | 0 | 1 | 0 | 2 |
| Total | 6 | 1 | 4 | 3 | 14 |

TEST CASE ANALYSIS

| Section | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Client Application | 10 | 0 | 3 | 7 |
| Security | 2 | 0 | 1 | 1 |
| Performance | 3 | 0 | 1 | 2 |
| Exception Reporting | 2 | 0 | 0 | 2 |

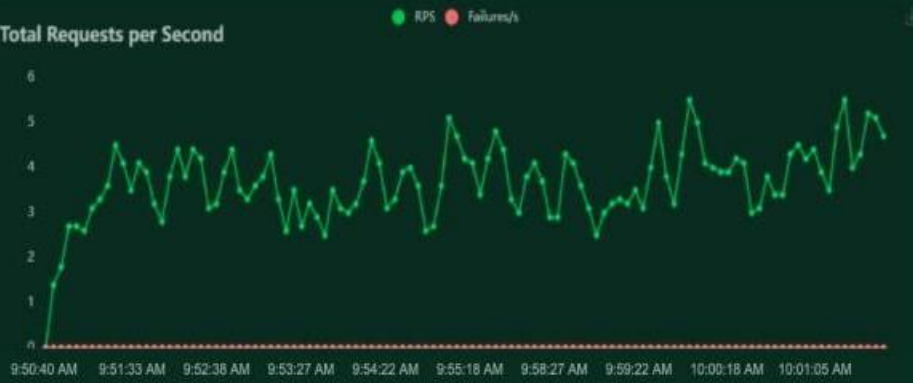
9. RESULTS

9.1 Performance Metrics

| Locust Test Report | | | | | | | | | |
|----------------------------------------------------------|----------|-------------|-------------|--------------|-------------|-------------|----------------------|-------------|--------------|
| During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM | | | | | | | | | |
| Target Host: http://127.0.0.1:5000/ | | | | | | | | | |
| Script: locust.py | | | | | | | | | |
| Request Statistics | | | | | | | | | |
| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
| GET | / | 1043 | 0 | 13 | 4 | 290 | 1079 | 1.9 | 0.0 |
| GET | /predict | 1005 | 0 | 39648 | 385 | 59814 | 2670 | 1.8 | 0.0 |
| Aggregated | | 2048 | 0 | 19462 | 4 | 59814 | 1859 | 3.7 | 0.0 |
| Response Time Statistics | | | | | | | | | |
| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
| GET | / | 10 | 11 | 13 | 15 | 19 | 22 | 62 | 290 |
| GET | /predict | 44000 | 46000 | 47000 | 48000 | 50000 | 52000 | 55000 | 60000 |
| Aggregated | | 36 | 36000 | 43000 | 45000 | 48000 | 50000 | 54000 | 60000 |

Charts

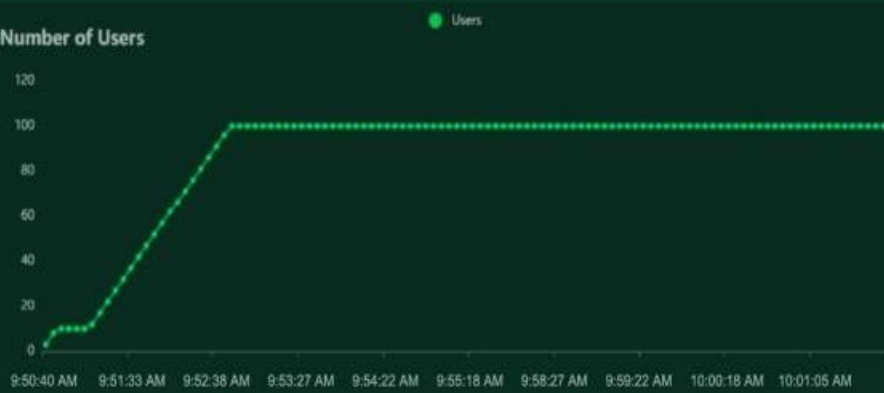
Total Requests per Second



Response Times (ms)



Number of Users



10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional error

11. CONCLUSION

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

12. FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement.

Some of the improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better.

Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

13. APPENDIX

SOURCE CODE

MODEL CREATION

Importing the required libraries

```
import numpy
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvolutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt #used for data visualization
```

[46]

load data

```
(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test
```

[47]

+ Code

+ Markdown

```
print (x_train.shape) #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)
```

[48]

```
... (60000, 28, 28)
     (10000, 28, 28)
```

```
x_train[0]
```

```
[49]
```

```
... Output exceeds the size limit. Open the full output data in a text editor
```

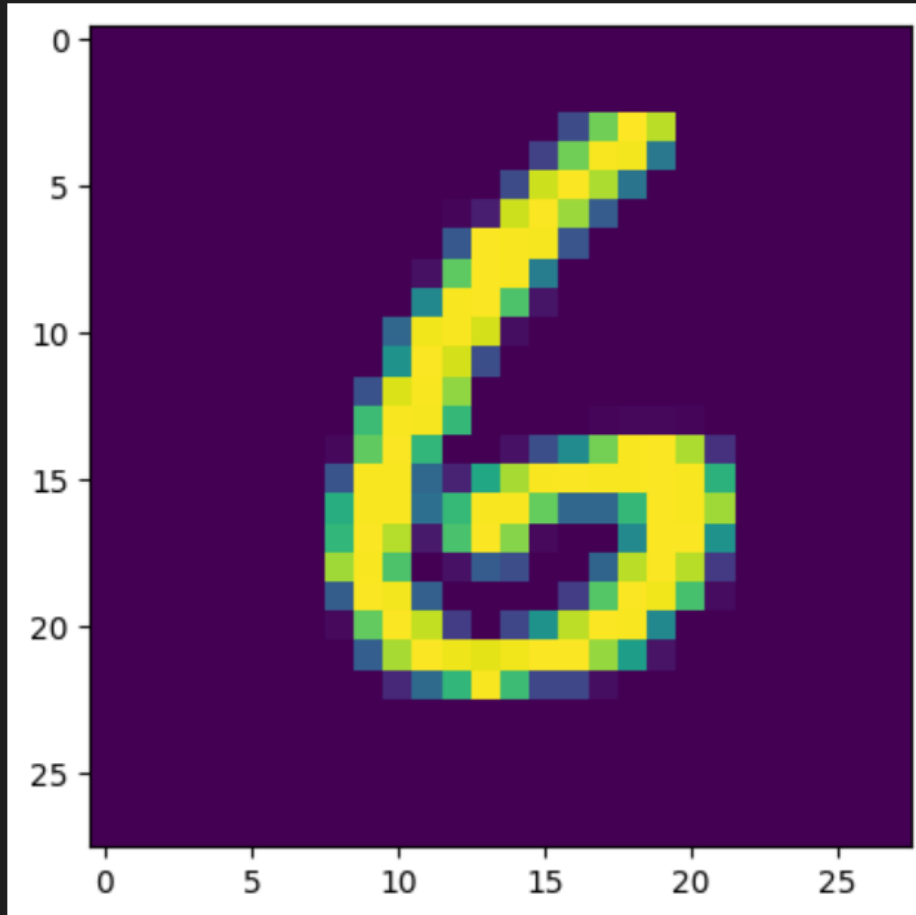
```
array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        3, 18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127,  0,  0,
        0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 30, 36, 94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195, 64,  0,  0,
```

```
plt.imshow(x_train[6000]) #ploting the index=image
```

[50]

```
... <matplotlib.image.AxesImage at 0x222c71e0250>
```

</>



▷ ▾

```
np.argmax(y_train[6000])
```

[51]

Reshaping Dataset

```
#Reshaping to format which CNN expects (batch, height, width, channels)
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

[52]

Applying One Hot Encoding

```
number_of_classes = 10 #storing the no of classes in a variable
```

[53]

```
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

[54]

Add CNN Layers

```
#create model
model=Sequential ()
```

[55]

```
#adding model Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

[56]

```
#flatten the dimension of the image
model.add(Flatten())
```

[57]

```
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

[58]

Compiling the model

```
#Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

[59]

```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

[62]

Train the model

```
#fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

[63]

```
Epoch 1/5
1875/1875 [=====] - 88s 47ms/step - loss: 0.2299 - accuracy: 0.9506 - val_loss: 0.0899 - val_accuracy: 0.9730
Epoch 2/5
1875/1875 [=====] - 81s 43ms/step - loss: 0.0682 - accuracy: 0.9796 - val_loss: 0.0765 - val_accuracy: 0.9780
Epoch 3/5
1875/1875 [=====] - 80s 43ms/step - loss: 0.0493 - accuracy: 0.9838 - val_loss: 0.0888 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [=====] - 80s 43ms/step - loss: 0.0357 - accuracy: 0.9887 - val_loss: 0.0835 - val_accuracy: 0.9791
Epoch 5/5
1875/1875 [=====] - 89s 48ms/step - loss: 0.0288 - accuracy: 0.9907 - val_loss: 0.1145 - val_accuracy: 0.9708

<keras.callbacks.History at 0x222d90d9db0>
```

Observing the metrics

```
# Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

[64]

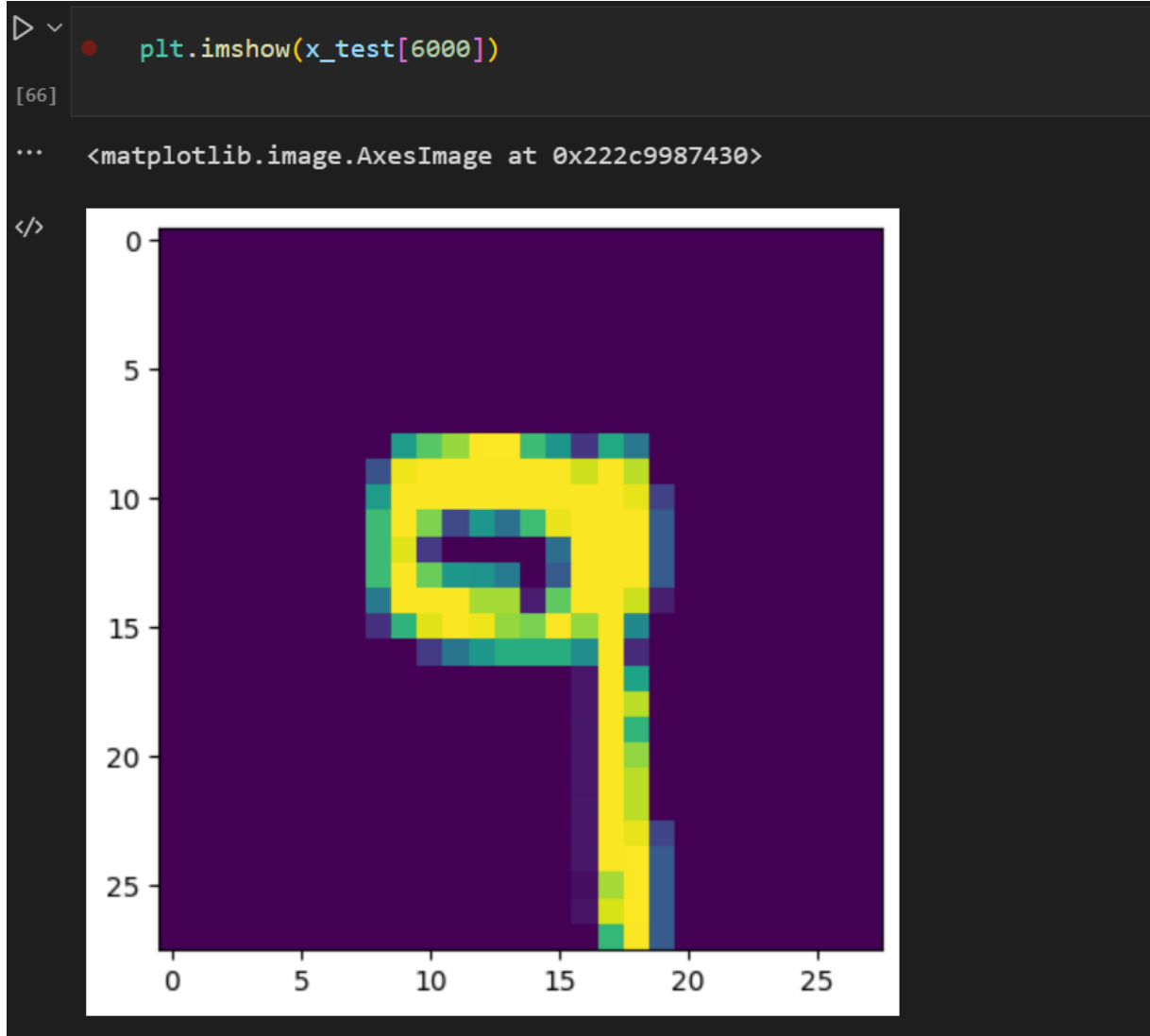
```
... Metrics (Test loss &Test Accuracy) :
      [0.1144733875989914, 0.97079998254776]
```

Test The Model

```
prediction=model.predict(x_test[6000:6001])
print(prediction)
```

[65]

```
... 1/1 [=====] - 0s 84ms/step
      [[1.4689527e-14 1.8748821e-17 2.3109615e-13 9.5624716e-07 3.1683821e-04
        2.4785629e-07 6.1842380e-18 6.2185841e-06 8.8746788e-07 9.9967492e-01]]
```



```
import numpy as np
print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images
```

[67]

... [9]

```
np.argmax(y_test[6000:6001]) #printing the actual labels
```

[68]

... 9

Save The model



```
# Save the model  
model.save('models/mnistCNN.h5')
```

[70]

FLASK APP

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from event.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = 'C:\\Users\\nandh\\IBM_NK\\Final Deliverables\\Final Code\\A-novel-method-for-digit-recognition-system\\flask_app\\uploads'

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("mnistCNN.h5")

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

```
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L") # convert image to monochrome
        img = img.resize((28, 28)) # resizing of input image

        im2arr = np.array(img) # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1) # printing our Labels

        return render_template('predict.html', num=str(num[0]))

if __name__ == '__main__':
    app.run(debug=False, threaded=False)
```

Index.html

```
<!DOCTYPE html>

<html lang="en">

<!DOCTYPE html>

<html lang="en">


<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="./index1.css">

</head>


<body>

    <div class="topnav" id="myTopnav">

        <a href="#" class="active">Home</a>

        <a href="./login.html">Login</a>

        <a href="./register.html">Register</a>

        <a href="./predict.html">predict</a>

        <p style="color: aquamarine; float: right; padding-right: 10px;"> Car Damage
Assissment</p>

    </div>

    <div class="contain_bar">

        <h3>

            Welcome to car Damage Assissment

        </h3>
```

<p>

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.

</p>

</div>

</body>

</html>

Index.css

```
html {  
    height: 100%;  
}  
  
body {  
    height: 100%;  
    margin: 0;  
    font-family: Arial, Helvetica, sans-serif;  
    display: grid;  
    justify-items: center;  
    align-items: center;  
    background-color: #3a3a3a;  
}  
  
#main-holder {  
    width: 50%;  
    height: 70%;  
    display: grid;  
    justify-items: center;  
    align-items: center;  
    background-color: white;  
    border-radius: 7px;  
    box-shadow: 0px 0px 5px 2px black;  
}
```

```
#login-error-msg-holder {  
  
    width: 100%;  
  
    height: 100%;  
  
    display: grid;  
  
    justify-items: center;  
  
    align-items: center;  
  
}  
  
#login-error-msg {  
  
    width: 23%;  
  
    text-align: center;  
  
    margin: 0;  
  
    padding: 5px;  
  
    font-size: 12px;  
  
    font-weight: bold;  
  
    color: #8a0000;  
  
    border: 1px solid #8a0000;  
  
    background-color: #e58f8f;  
  
    opacity: 0;  
  
}  
  
#error-msg-second-line {  
  
    display: block;  
  
}  
  
#login-form {  
  
    align-self: flex-start;  
  
    display: grid;
```

```
    justify-items: center;

    align-items: center;

    color: #3a3a3a;
}

.login-form-field::placeholder {

    color: #3a3a3a;
}

.login-form-field {

    border: none;

    border-bottom: 1px solid #3a3a3a;

    margin-bottom: 10px;

    border-radius: 3px;

    outline: none;

    padding: 0px 0px 5px 5px;
}

#login-form-submit {

    width: 100%;

    padding: 7px;

    border: none;

    border-radius: 5px;

    color: white;

    font-weight: bold;

    background-color: #3a3a3a;

    cursor: pointer;

    outline: none;
```

```
}
```

```
.topnav {  
    background-color: #333;  
    overflow: hidden;  
}
```

```
.topnav a {  
    float: left;  
    display: block;  
    color: aqua;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
    font-size: 17px;  
}
```

```
.topnav a:hover {  
    background-color: aqua;  
    color: black;  
    border-radius: 30%;  
}
```

```
.topnav a.active {  
    background-color: royalblue;  
    color: white;  
}
```



```
.topnav .icon {  
  display: none;  
}  
  
.contain_bar {  
  position: relative;  
  top: 10px;  
  height: 300px;  
  padding: 100px;  
  border: greenyellow 2px solid;  
  display: block;  
  overflow: hidden;  
}
```

Predict.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="./index1.css">

  <link rel="stylesheet" href="./login-page.css">

</head>

<body>

  <div class="topnav" id="myTopnav">

    <a href="./indextsajid.html" >Home</a>

    <a href="./login.html">Login</a>

    <a href="./register.html" >Register</a>

    <!-- <a href="./predict.html" class="active">predict</a> -->

    <p style="color: aquamarine; float: right; padding-right: 10px;"> Car Damage Assissment</p>

  </div>

  <div id="page">

    <div id="content">

      <div style="margin-bottom: 20px;">

        <p align="justify"><strong>

          <form action="/predict" method="post" enctype="multipart/form-data" name="form1">

            <table border="0" align="center">

              <tr>
```



```

        <td>&nbsp;</td>

        <td><label>

            <input type="submit" name="Submit" value="Submit" />

            <input type="reset" name="Submit2" value="Reset">

            <a href="user_reg.jsp"></a></label></td>

    </tr>

</table>

</form>

</p>

<p>&nbsp;</p>

</blockquote>

</div>

<div>&nbsp;</div>

<div class="twocols"></div>

</div>

<div>

    <button class="btn btn-danger" style="float: right;" onclick="logout()">Log out</button>

    <script src="./logout.js"></script>

</div>

<!-- end content -->

<!-- end sidebar -->

<div style="clear: both;">&nbsp;</div>

</div>

</body>

</html>

```

Main.py

```
+from flask import Flask, render_template, flash, request, session
from cloudant.client import Cloudant

import cv2

client = Cloudant.iam(08a0c161-7d82-42e7-9ac7-ebea49405c67--
bluemix", "Ou7wZe3SAsX6TepUpwKdDANQiYxePUTY_QXsgmxgl5j
my_database = client.create_database("database-my_database")

app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

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

    return render_template('indextsajid.html')

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

    return render_template('login.html')
@app.route("/addamount")

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

    return render_template('registration.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)
```


Git hub link : <https://github.com/IBM-EPBL/IBM-Project-50161-1660897196>

Project Demo link :

<https://drive.google.com/drive/folders/1yKXFlkSDNhQhYUmsPwsZDGdQ0yJNMJ5T?usp=sharing>