INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES USING IBM CLOUD

INDUSTRY ORIENTED MINI PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,

HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MANTHRI MANIKANTA	**19UK1A05E9**
SYED ALI SAEED BAQRI	**19UK1A05F9**
PORIKA ANUPRIYA	**19UK1A05H3**

Under the esteemed guidance of

Ms. K. SOUMYA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
Bollikunta, Warangal – 506005
2019– 2023

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED MINI PROJECT

This is to certify that the Industry Oriented Mini Project entitled "**INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES USING IBM CLOUD"** is being submitted by ***M. MANIKANTA(H.NO:19UK1A05E9), SYED ALI SAEED BAQRI(H.NO:19UK1A05F9), P. ANUPRIYA(H.NO:19UK1A05H3)*** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2021-2022** , is a record of work carried out by them under the guidance and supervision.

Project Guide
K. SOUMYA
(Assistant Professor)

Head of the Department
DR. R. NAVEEN KUMAR
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this in the Industry Oriented Mini Project in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the Industry Oriented Mini Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the Industry Oriented Mini Project and for their support in completing the Industry Oriented Mini Project.

We express heartfelt thanks to the guide, **Ms. K. SOUMYA** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this Industry Oriented Mini Project.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

M.MANIKANTA (19UK1A05E9)
SYED ALI SAEED BAQARI (19UK1A05F9)
P.ANUPRIYA (19UK1A05H3)

ABSTRACT

Nowadays, the proliferation of automobile industries is directly related to the number of claims in insurance companies. Those companies are facing many simultaneous claims and solving claims leakage. In Advanced Artificial Intelligence (AI), machine learning and deep learning algorithms can help to solve these kinds of problems for insurance industries. In this project, we apply deep learning-based algorithm, VGG16 for car damage detection and assessment in real world datasets. The algorithm detect the damaged part of a car, assess its severity. Initially, we discover the effect of domain specific pre-trained CNN models, which are trained on using dataset. Then we apply transfer learning in pre-trained VGG models and use some techniques to improve the accuracy of our system. After analyzing and implementing our models, we can find out the Cost and we can estimate it results of a particular damage of vehicle.

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 technique that compares before-and after-accident car images to automatically detect the damaged location.

Keywords — Convolution Neural Network, Deep Learning, Image classification, R-CNN and object detection.

CONTENTS

S NO: TOPIC

1. INTRODUCTION

1.1 Overview

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.

1.2 Background

Currently, after a vehicle has been damaged in a road accident or otherwise, the vehicle must be taken by the owner or a tow company to an auto repair shop for inspection. Inspection of the vehicle by a mechanic at the auto repair shop is required in order to assess which parts of the vehicle need to be repaired or replaced. An estimate is then generated based on the inspection. In some cases, when an insurance claim is filed, the estimate is forwarded to an insurance company to approve the repairs before the repairs are made to the vehicle.

From end-to-end, the process of vehicle inspection, estimate generation, claim approval, and vehicle repair can be long and complex, involving several parties including at least a customer, an auto repair shop, and a claim adjustor.

1.3 Summary

One embodiment of the disclosure includes a method for automatically estimating a repair cost for a vehicle, comprising: receiving, at a server computing device over an electronic network, one or more images of a damaged vehicle from a client computing device; performing computerized image processing on each of the one or more images to detect damage to a set of parts of the vehicle; and, calculating an estimated repair cost for the vehicle based on the detected damage based on accessing a parts database that includes repair costs. Additionally, in some embodiments, the server computing device may classify the loss as a total, medium, or small loss.

2. LITERATURE SURVEY

2.1 Problem Statement

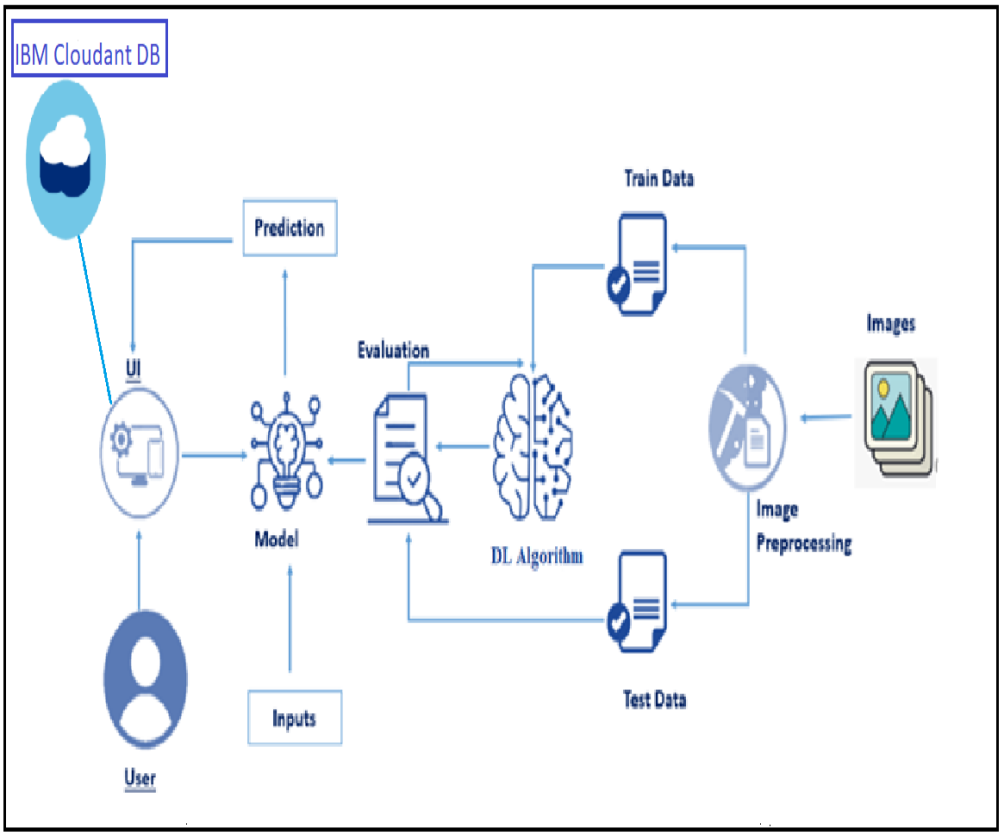
Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims.

2.2 Proposed Solution

In this project I am going 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.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



The above block diagram tells about the working process of Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies using IBM Cloud.

3.2 Prerequisite:

3.2.1 Software Requirements:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum, and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio and Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter

Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Python specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Web framework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

3.2.2 Hardware Requirements:

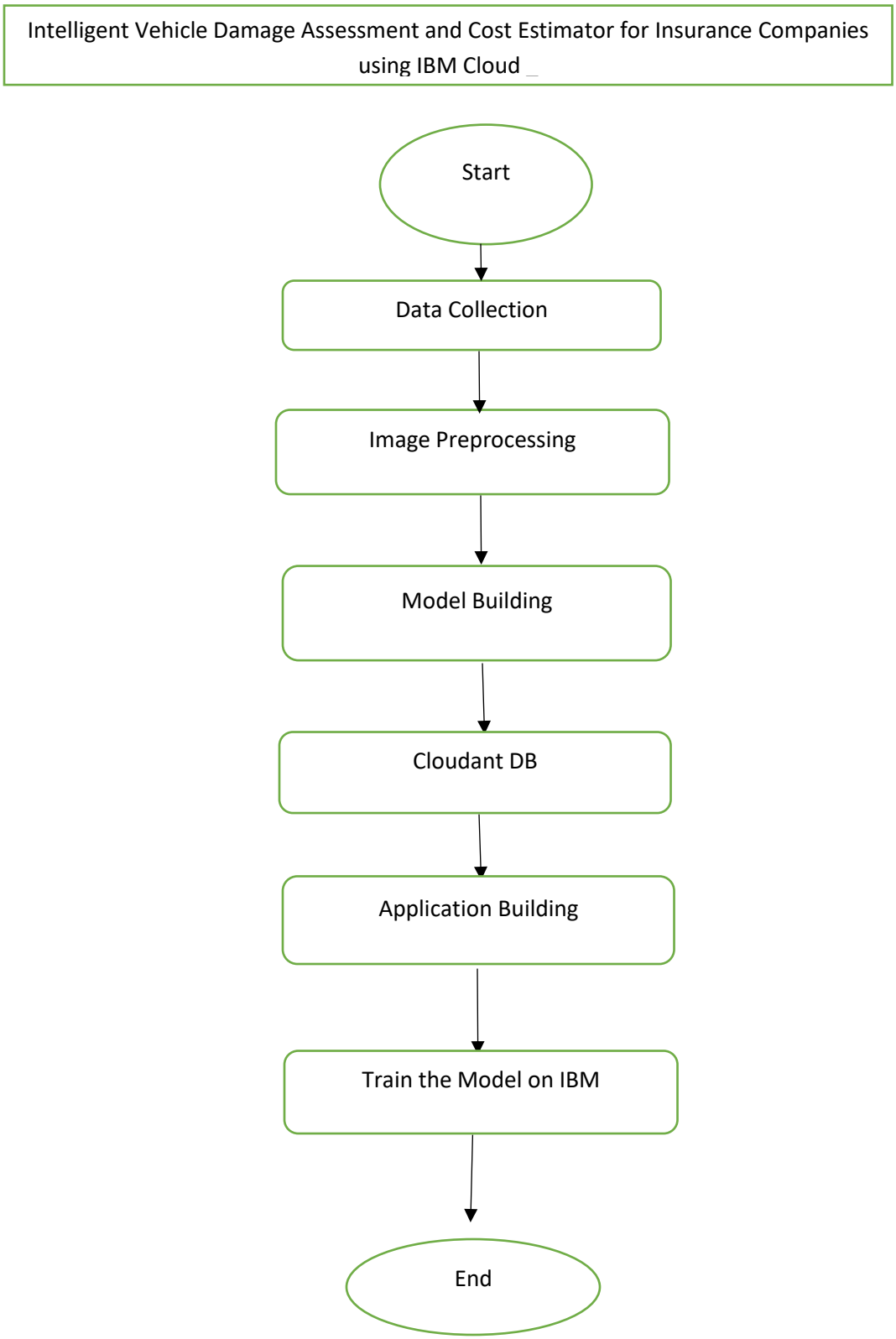
- Operating system: window 7 and above with 64bit
- Processor Type -Intel Core i3-3220
- RAM: 4Gb and above
- Hard disk: min 100GB

4. EXPERIMENTAL INVESTIGATIONS

Coming to analysis or investigations VGG16 model is selected for this problems So the Car Damage is identified by using this vgg16 model. There are three situations to consider for a car damaged level. According to Libertymutual.com, the classification of damages are as follows Minor Damage - scratches headlight or small dent in hood of a car. Moderate Damage - large dents in hood, fender or door of a car. Severe Damage - broken axes, bent or twisted frames and destroy air bags of a car.VGG16 64 feature kernel filters are used by the first and second convolutional layers. • The third and fourth of convolutional layers are 124 feature kernel filters and the output will be decreased the input size of 224 × 224× 3 into 56 × 56× 128. The convolutional layers from fifth to eighth use 256 feature maps. 512 filters are used by the two sets of convolutional layers from ninth to sixteen. The full connected hidden layers of seventeen and eighteen layers have 4,096 nodes with ReLU and the last one has 1,000 nodes with softmax.

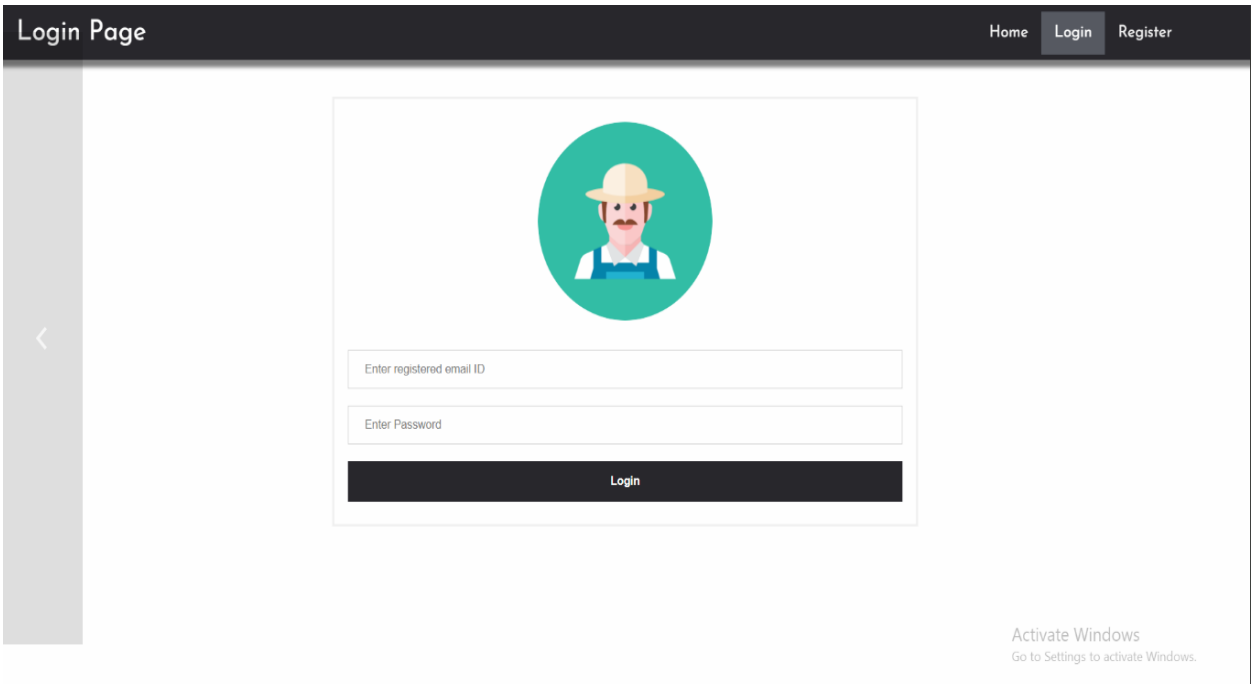
5. FLOWCHART

In the below flowchart, the working process of Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies using IBM Cloud is showed.

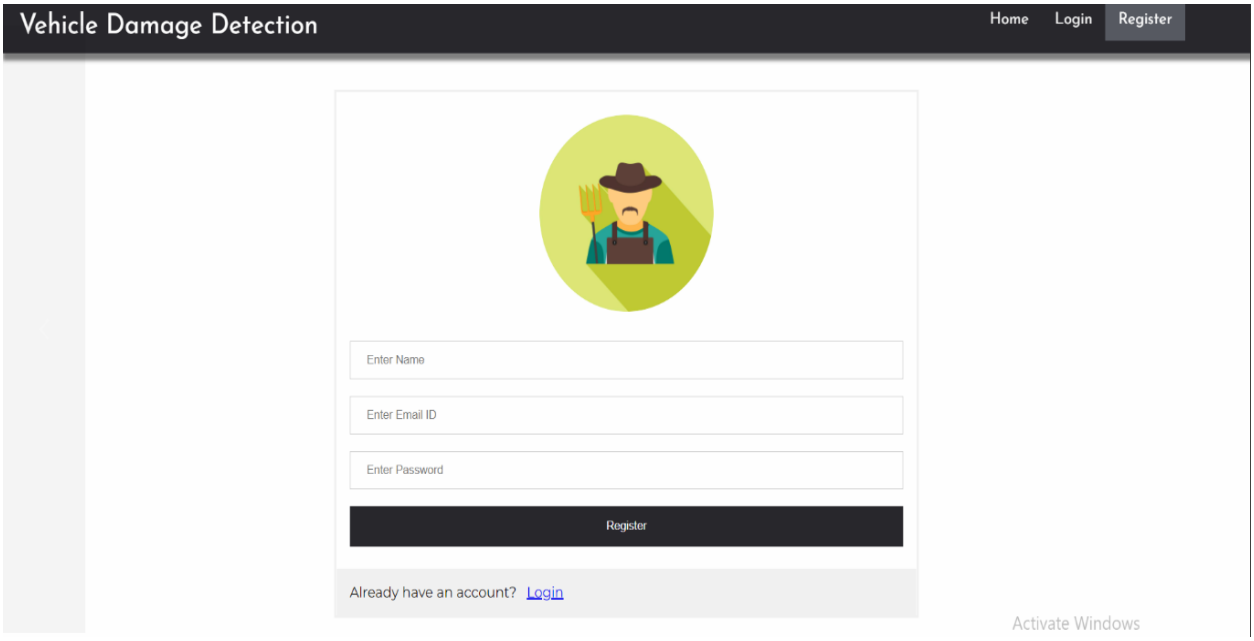


6. RESULT

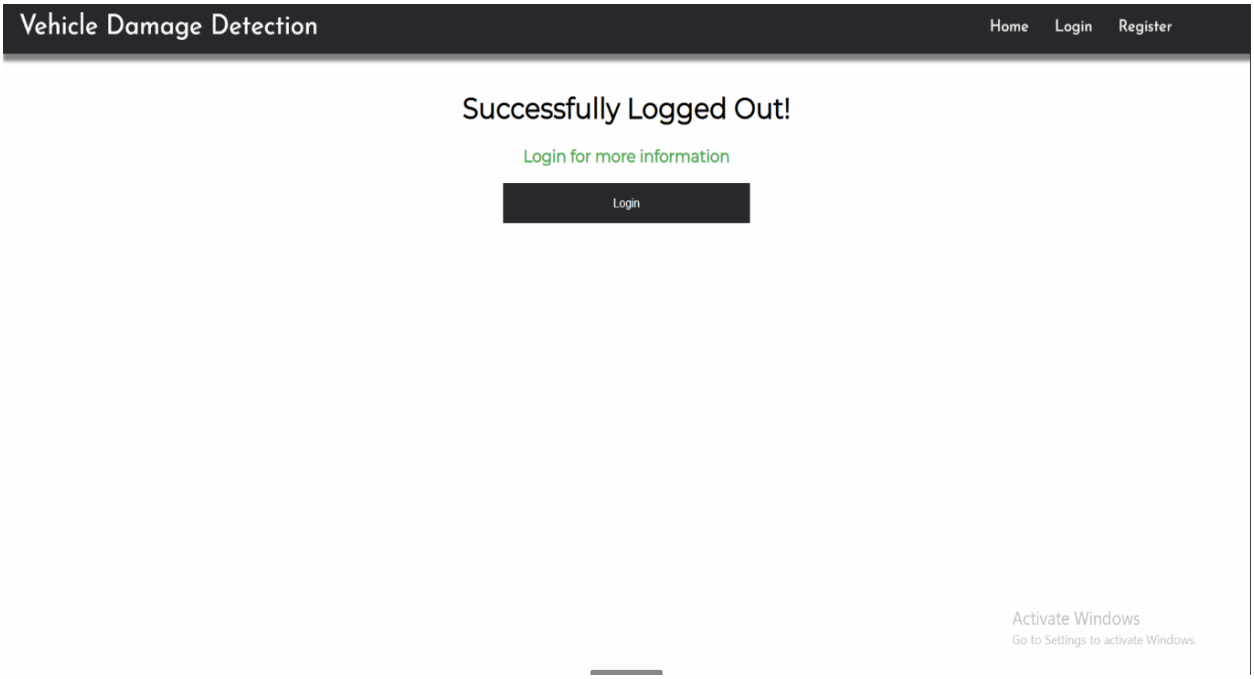
Login page:-



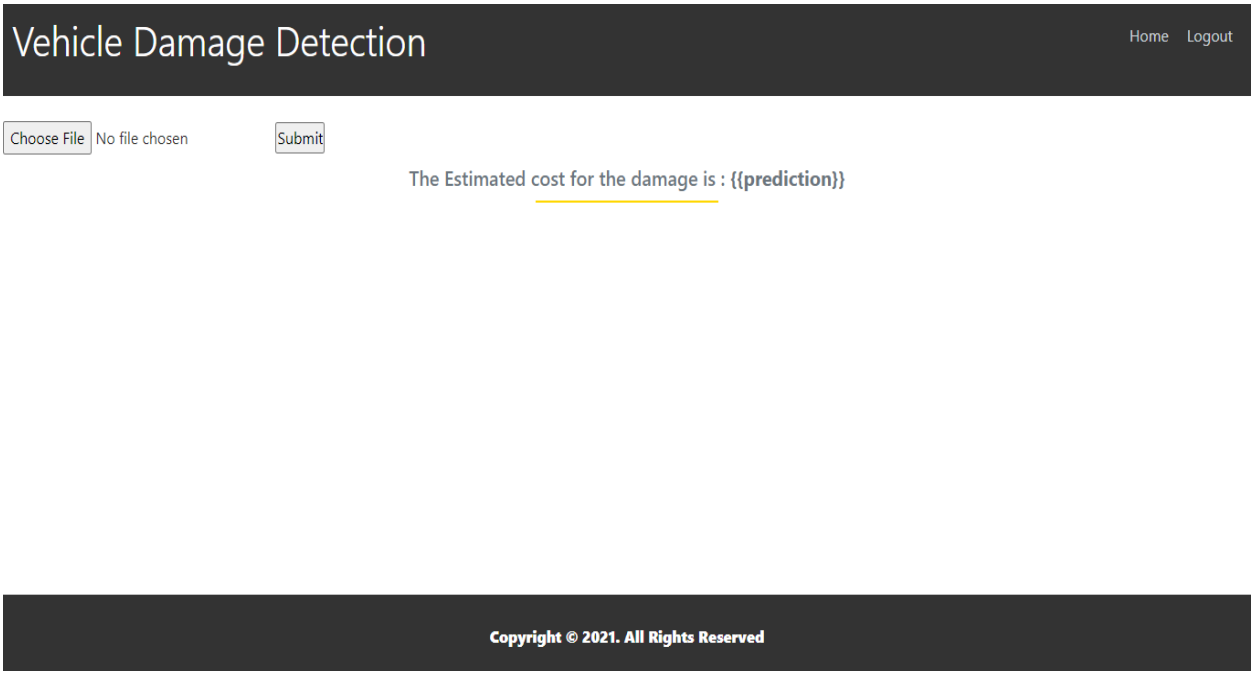
Register page:-



Logout:-



Prediction page:



7. ADVANTAGES and DISADVANTAGES

ADVANTAGES:-

- It can categorize the proportion of damaged parts and determine whether they need to be replaced or repaired.
- It aids the user in expediting the process of filing an insurance claim for his vehicle.
- Get a report with the vehicle's damage analysis created for you. To get compensation, submit the created report. Process that saves time and money.
- Efficient program for detecting damages on car.
- Works intelligently to estimate costs for the damages.
- Makes processing of claims faster for the insurance companies.

DISADVANTAGES:

- Privacy
- Low accuracy for complex data
- Process complicated and unstable result

8. APPLICATIONS

This application can further be developed with more idea and implementation and by using different algorithms. The accuracy cost of the model can be further improved by using VGG16 of CNN algorithm which is used to predict the damage cost of vehicle for insurance companies of leakage claims .It proposes to improve the accuracy further.

9. CONCLUSION AND FUTURE SCOPE

Using this project, we can detect the area of damage on car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if the users can upload pictures and the model can assess damage.

We described applicable deep learning-based algorithms for car damage assessment in the real world datasets. We created new datasets when there is no openly obtainable dataset for car damage classification. What is more, we experimented with the deep learning-based pre trained VGG16 models from random initialization.

In this work of Damage analysis of a vehicle in general and insurance reclaim, a system has been designed using CNN and image classification which takes the input from a user as an image to test the severity of damage, which happens in a sequence of two steps. First being the image classification, here the input provided by the user is processed by the neural network to identify the car that is if the car is damaged or not. And later on the second step, the flattened input obtained as the output in step 1 is applied for object detection to identify the region and severity of damage, where region might be rear, front or side and severity is divided into minor, moderate and major. The R-CNN network identifies the severity of damage and a report is filed and sent to the user and the insurance firm.

The major drawback of the proposed model is that it only identifies the physical visible damage and not of the internal or the interior damage. These models followed by supervised fine tuning and transfer learning with L2 regularization technique to fit our specific task. We observed that training with a small dataset is not sufficient to get the best accuracy based on deep learning approach. In addition to this, it was not enough to use just one of the regularization technique in our

system. After analyzing our models, we find out that the results of using transfer learning and regularization can work better than those of fine-tuning. All of the above, our pre-trained models not only detect damaged part but also assess its location and severity.

That’s why this solution can help the asset for insurance companies to solve claims leakage problems. Regarding to our proposed models, we still face the over fitting problem in our models. Thus, in future work, we need to utilize other types of regularization techniques with a large dataset. If we have higher quality datasets, including the features of a car (make, model and the year of manufacture), location information, type of damaged part and repair cost, we can predict the cost of a car damaged part to be more reliable and accurate.

11. BIBLIOGRAPHY

We referred some books and searched in the internet for the better outcome of the project:

[1] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, ”Extreme gradient boosting machine learning algorithm for safe auto insurance operations,” in IEEE International Conference on Vehicular Electronics and Safety (ICVES’19), Cairo, Egypt, Sept. 2019.
[2] M. Wassel, ”Property Casualty: Deterring Claims Leakage in the Digital Age,” Cognizant Insurance Practice, Tech. Rep., 2018.
[3] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, ”A Very Deep Transfer Learning Model for Vehicle Damage Detection and Localization,” in IEEE International Conference on Vehicular Electronics and Safety (ICVES’19), Cairo, Egypt, Sept. 2019.
[4] K. Patil, M. Kulkarni, A. Sriraman and S. Kerande, ”Deep Learning Based Car Damage Classification,” in IEEE Int. Conf. Machine Learning App. (ICMLA’17), Cancun, Mexico, Dec. 2017.
[5] j. d. Deijin, ”Automatic Car Damage Recognition using Convolutional Neural Networks,” March 2018

12. APPENDIX

A COLAB NOTEBOOK

https://colab.research.google.com**/drive/1fZVJ8x8Yy81aeatAhZm9xS65-HEbthRR?usp=share_link**

FLASK CODE:

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

from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam("eb287972-7219-4709-93ef-1837a26cadf8-
bluemix", "Efge5UTUz85XxZ8QNAYb65N3kQFJwgep_PgEGyF5pr2e",connect=True)


# Create a database using an initialized client
my_database = client.create_database('my_database')


model1=load_model("body.h5")
model2=load_model("level.h5")

app=Flask(__name__)

#default home page or route
@app.route('/')
```

```
def index():
    return render_template('index.html')



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


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

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

    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        url = my_database.create_document(data)
        #response = requests.get(url)
        return render_template('register.html', pred="Registration Success
ful, please login using your details")
    else:
        return render_template('register.html', pred="You are already a me
mber, please login using your details")

#login page
@app.route('/login')
def login():
```

```
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))


    if(len(docs.all())==0):
        return render_template('login.html', pred="The username is not fou
nd.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')


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

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


@app.route('/result',methods=["GET","POST"])
def res():
    if request.method=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__) #getting the current path i.e w
here app.py is present
        #print("current path",basepath)
        filepath=os.path.join(basepath,'uploads',f.filename) #from anywher
e in the system we can give image but we want that image later  to process
 so we are saving it to uploads folder for reusing
```

```
        #print("upload folder is",filepath)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(224,224))
        x=image.img_to_array(img)#img to array
        x=np.expand_dims(x,axis=0)#used for adding one more dimension
        #print(x)
        img_data=preprocess_input(x)
        print(model1.predict(img_data), model2.predict(img_data))
        prediction1=np.argmax(model1.predict(img_data))
        prediction2=np.argmax(model2.predict(img_data))
        print(prediction1, prediction2)
        #prediction=model.predict(x)#instead of predict_classes(x) we can
use predict(X) ---->predict_classes(x) gave error
        #print("prediction is ",prediction)
        index1=['front', 'rear', 'side']
        index2=['minor', 'moderate', 'severe']
        #result = str(index[output[0]])
        result1 = index1[prediction1]
        result2 = index2[prediction2]
        print(result1,result2)
        if(result1 == "front" and result2 == "minor"):
            number = "3000 - 5000 INR"

        elif(result1 == "front" and result2 == "moderate"):
            number = "6000 - 8000 INR"

        elif(result1 == "front" and result2 == "severe"):
            number = "9000 - 11000 INR"

        elif(result1 == "rear" and result2 == "minor"):
            number = "4000 - 6000 INR"

        elif(result1 == "rear" and result2 == "moderate"):
            number = "7000 - 10000 INR"

        elif(result1 == "rear" and result2 == "severe"):
            number = "11000 - 13000 INR"

        elif(result1 == "side" and result2 == "minor"):
             number = "6000 - 8000 INR"

        elif(result1 == "side" and result2 == "moderate"):
             number = "9000 - 11000 INR"
```

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

        else:
            number = "15000 - 50000 INR"

        return render_template('prediction.html', prediction=number)
```