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
SATHARLA SRUTH	19UK1A05E6
PORIKA ANUPRIYA	19UK1A05H3

Under the esteemed guidance of

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),S. SRUTHI (H.NO:19UK1A05E6), 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 **2021-22**, is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
K. SOUMYA	**DR. R. NAVEEN KUMAR**
(Assistant Professor)	(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, **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)
S.SRUTHI	(19UK1A05E6)
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 pretrained 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.

CONTENTS

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.

Accordingly, there is a need in the art for an improved system that overcomes some of the drawbacks and limitations of conventional approaches.

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 Existing Problem (OR) 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
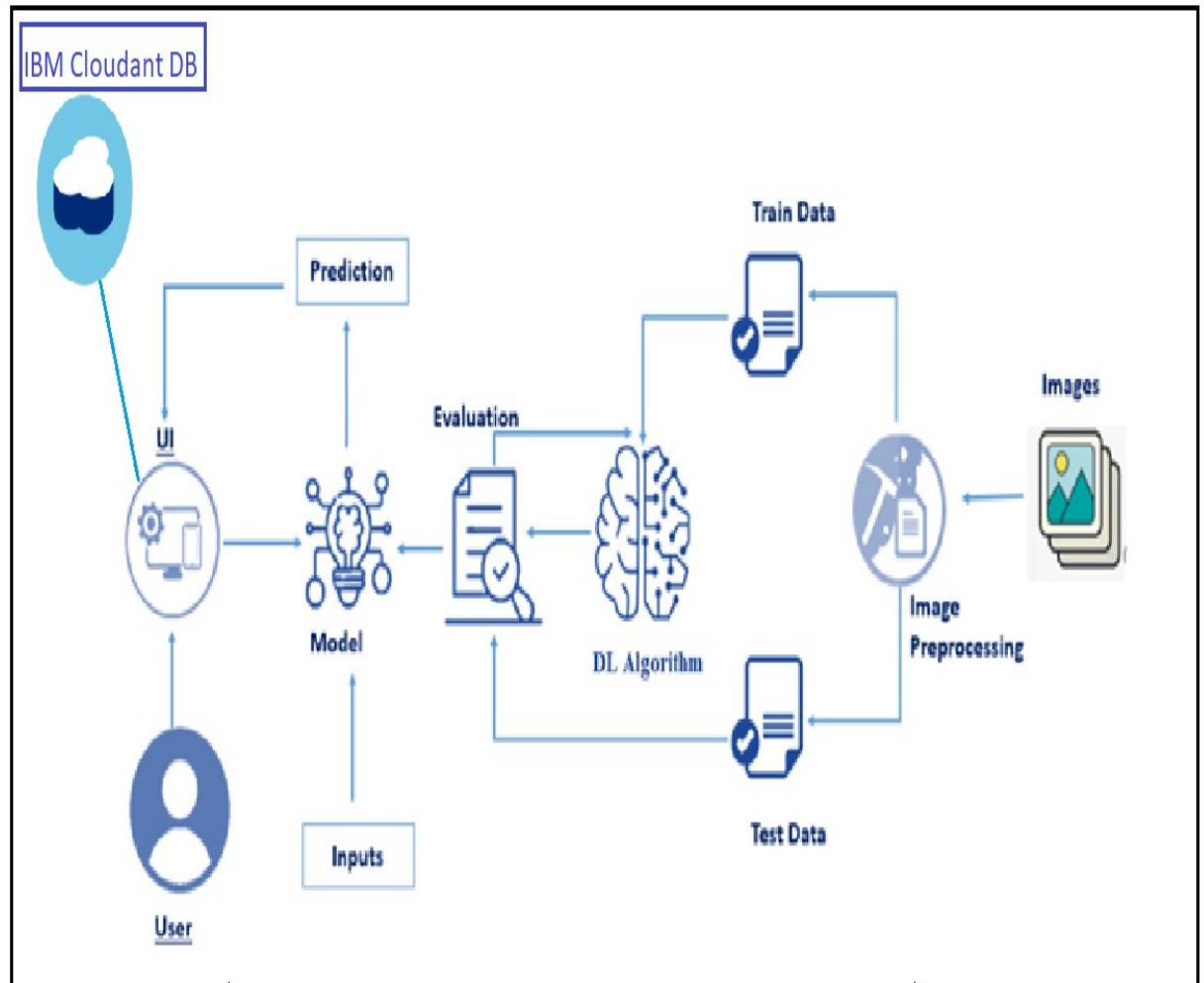
Prerequisites:

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 Pythonspecific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Webframework 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.

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 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

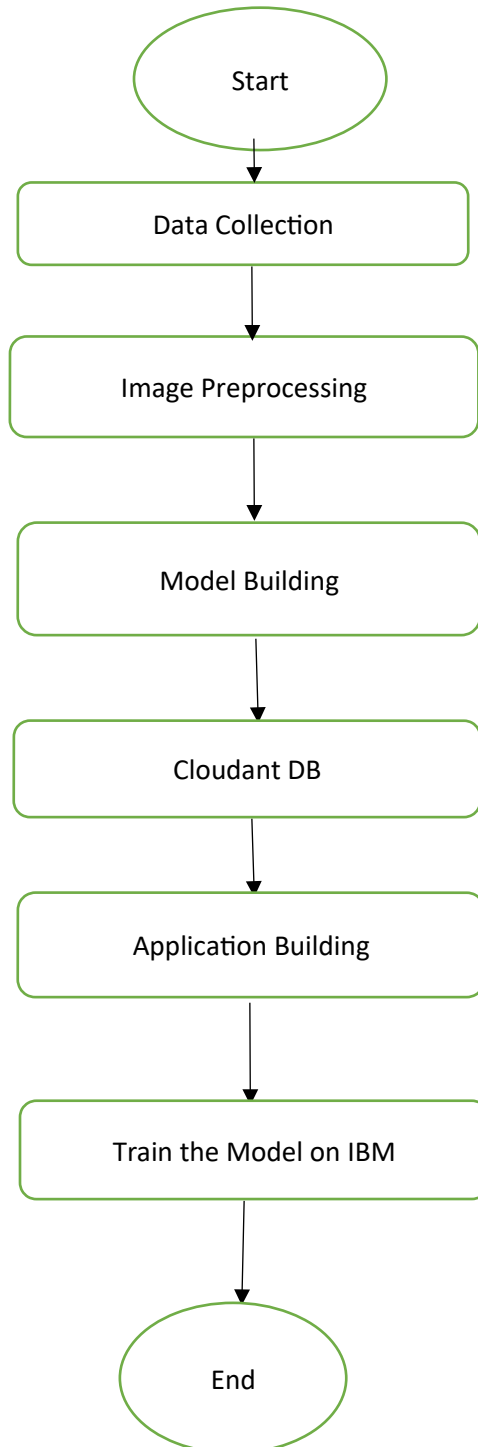Minor Damage - scratches headlight or small dent in hood of a car.

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 $224 \times 224 \times 3$ into $56 \times 56 \times 128$. • 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

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies using IBM Cloud _

Start

Data Collection

Image Preprocessing

Model Building

Cloudant DB

Application Building

Train the Model on IBM

End

6.RESULT

Login page:-

Register page:-

Logout:-

Vehicle Damage Detection   Home   Login   Register

Successfully Logged Out!

Login for more information

Login

Prediction page:

Vehicle Damage Detection   Home   Logout

Choose File   No file chosen   Submit

The Estimated cost for the damage is : {{prediction}}
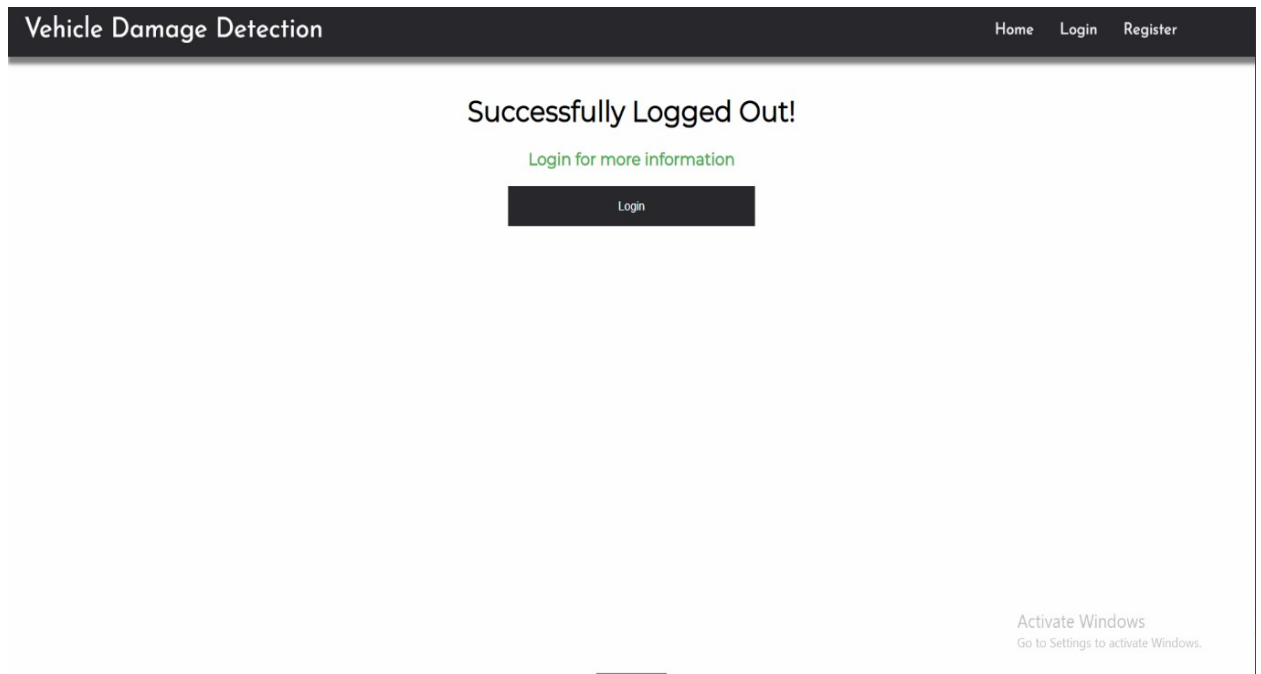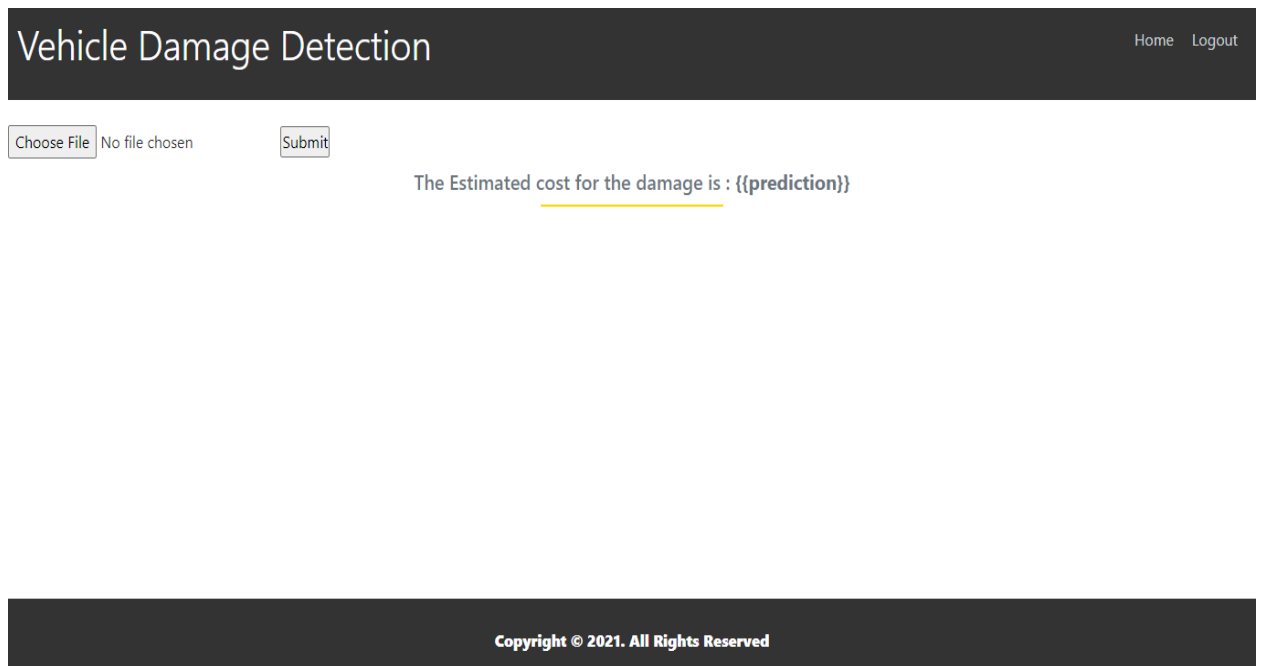
7.ADVANTAGES and DISADVANTAGES

ADVANTAGES:

- 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

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.

10.FUTURE SCOPE

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 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:

```python
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", "Efg
e5UTUz85XxZ8QNAYb65N3kQFJwgep_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('/')
```

```python
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():
```

```python
    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
```

```python
        #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)
```

HTML CODE

index.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">

<!--Bootstrap -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

```html
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>


<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap"
rel="stylesheet">

<link rel="stylesheet" href="../static/style.css">

<!-- <script defer src="../static/js/main.js"></script> -->

<style>

</style>

<title>Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies using
DL`</title>

</head>

<body>

<header id="head" class="header">

<section id="navbar">

<h3 class="nav-heading"></i>Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance
Companies</h3>

<div class="nav--items">

<ul>

<li><a href="{{ url_for('index')}}">Home</a></li>

<li><a href="{{ url_for('login')}}">Login</a></li>

<li><a href="{{ url_for('register')}}">Register</a></li>

<!-- <li><a href="#about">About</a></li>
```

```html
<li><a href="#services">Services</a></li> -->

<li><a href='/prediction'>Prediction</a></li>

</ul>

</div>

</section>


</header>

<section id="about">

<div class="top">

<h3 class="title text-muted">

ABOUT PROJECT

</h3>

<div class="line"></div>

</div>

<div class="body", style="white-space:pre">



<p style="font-size:150%;">

Vehicle damage detection is used to reduce claims leakage during insurance processing.

Visual inception and validation are usually done. As it takes a long time, because a person needs to come and inspect the damage.

Here we are trying to automate the procedure. Using this automation, we can avoid time conception for the insurance claim procedure.

</p>
```

```html
</div>

</section>



<section id="footer">

<p>Copyright © 2021. All Rights Reserved</p>

<div class="social">

<a href="#" target="_blank"><i class="fab fa-2x fa-twitter-square"></i></a>

<a href="#" target="_blank">

<i class="fab fa-2x fa-linkedin"></i></a>

<a href="#">

<i class="#"></i>

</a>

</div>

</section>

</body>

</html>
```

login.html


```html
<!DOCTYPE html>

<html >
```

```html
<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Login Page</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<!link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>


<style>

.header {

top:0;

margin:0px;

left: 0px;

right: 0px;

position: fixed;

background-color: #28272c;

color: white;
```

```
box-shadow: 0px 8px 4px grey;

overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: center;

}

.topnav {

overflow: hidden;

background-color: #333;

}


.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 18px;

}


.topnav-right a:hover {
```

```
background-color: #ddd;

color: black;

}


.topnav-right a.active {

background-color: #565961;

color: white;

}


.topnav-right {

float: right;

padding-right:100px;

}


.login{

margin-top:-70px;

}

body {


background-color:#ffffff;

background-repeat: no-repeat;

background-size:cover;

background-position: 0px 0px;

}
```

```css
.login{

margin-top:100px;

}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] {

width: 100%;

padding: 12px 20px;

display: inline-block;

margin-bottom:18px;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #28272c;

color: white;

padding: 14px 20px;

margin-bottom:8px;

border: none;

cursor: pointer;

width: 100%;

font-weight:bold;
```

```css
}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}

img.avatar {

width: 30%;

border-radius: 50%;

}

.container {

padding: 16px;
```

```css
}

span.psw {

float: right;

padding-top: 16px;


}


/* Change styles for span and cancel button on extra small screens */

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}
```

```html
</style>

</head>


<body style="font-family:Montserrat;">


<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Login
```

Page</div>

<div class="topnav-right" style="padding-top:0.5%;">

Home

Login

Register

</div>

</div>

<div id="login" class="login">

<form action="{{url_for('afterlogin')}}" method="post">

<div class="imgcontainer">

<img style="" src="https://pngimage.net/wp-content/uploads/2018/05/agriculteur-png.png" alt="Avatar" class="avatar">

</div>

<div class="container">

<input type="email" placeholder="Enter registered email ID" name="_id" required>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Login</button>

</div>

</form>

</div>

        </body>

        </html>

logout.html

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Vehicle Damage Detection for Insurance Companies</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<style>

.header {

top:0;

margin:0px;

left: 0px;

right: 0px;

position: fixed;

background-color: #28272c;

color: white;

box-shadow: 0px 8px 4px grey;

overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: center;

}

.topnav {

```css
overflow: hidden;

background-color: #333;

}


.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 18px;

}


.topnav-right a:hover {

background-color: #ddd;

color: black;

}


.topnav-right a.active {

background-color: #565961;

color: white;

}


.topnav-right {

float: right;
```

```css
padding-right:100px;

}

.login{

margin-top:-70px;

}

body {

background-color:#ffffff;

background-repeat: no-repeat;

background-size:cover;

background-position: 0px 0px;

}

.main{

margin-top:100px;

text-align:center;

}

form { margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] {

width: 100%;

padding: 12px 20px;

display: inline-block;
```

```css
margin-bottom:18px;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #28272c;

color: white;

padding: 14px 20px;

margin-bottom:8px;

border: none;

cursor: pointer;

width: 20%;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}
```

```css
.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}


img.avatar {

width: 30%;

border-radius: 50%;

}


.container {

padding: 16px;

}


span.psw {

float: right;

padding-top: 16px;

}


/* Change styles for span and cancel button on extra small screens */

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;
```

```
}

.cancelbtn {

width: 100%;

}

}


</style>

</head>


<body style="font-family:Montserrat;">


<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Vehicle
Damage Detection</div>

<div class="topnav-right" style="padding-top:0.5%;">


<a href="{{ url_for('home')}}">Home</a>

<a href="{{ url_for('login')}}">Login</a>

<a href="{{ url_for('register')}}">Register</a>

</div>

</div>

<div class="main">

<h1>Successfully Logged Out!</h1>

<h3 style="color:#4CAF50">Login for more information<h3>
```

```html
<a href="{{ url_for('login') }}"><button type="submit">Login</button></a>

</form>

</div>


</body>

</html>
```

prediction.html

```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">

<!--Bootstrap -->

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
```

crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../static/style.css">

<script defer src="../static/js/JScript.js"></script>

<title>Prediction</title>

</head>

<body>

<header id="head" class="header">

<section id="navbar">

<h1 class="nav-heading"></i>Vehicle Damage Detection</h1>

<div class="nav--items">

Home

Logout

<!-- About

Services -->


```html
</div>

</section>

</header>

<!-- dataset/Training/metal/metal326.jpg -->

<section id="prediction">

<div class="prediction-input">

</div>

<form id="form" action="/result" method="post" enctype="multipart/form-data">

<input type="file" id="imageupload" name="image" accept="image/*" class="input-image">

<input type="submit" class="submitbtn">

</form>

</div>

<h5 class="title text-muted">

The Estimated cost for the damage is : <b>{{prediction}}<b>

</h5>


<div class="line"></div>


</section>

<section id="footer">

<p>Copyright © 2021. All Rights Reserved</p>


</section>
```

</body>

</html>

register.html

```html
<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Vehicle Damage Detection</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">


<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
```

```
<style>

.header {

top:0;

margin:0px;

left: 0px;

right: 0px;

position: fixed;

background-color: #28272c;

color: white;

box-shadow: 0px 8px 4px grey;

overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: center;

}

.topnav {

overflow: hidden;

background-color: #333;

}
```

```css
.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 18px;

}


.topnav-right a:hover {

background-color: #ddd;

color: black;

}


.topnav-right a.active {

background-color: #565961;

color: white;

}


.topnav-right {

float: right;

padding-right:100px;

}
```

```css
.login{

margin-top:-70px;

}

body {

background-color:#ffffff;

background-repeat: no-repeat;

background-size:cover;

background-position: 0px 0px;

}

.login{

margin-top:100px;

}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] {

width: 100%;

padding: 12px 20px;

display: inline-block;

margin-bottom:18px;

border: 1px solid #ccc;

box-sizing: border-box;

}
```

```css
button {

background-color: #28272c;

color: white;

padding: 14px 20px;

margin-bottom:8px;

border: none;

cursor: pointer;

width: 100%;

}


button:hover {

opacity: 0.8;

}


.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}


.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}
```

```css
img.avatar {

width: 30%;

border-radius: 50%;

}


.container {

padding: 16px;

}


span.psw {

float: right;

padding-top: 16px;

}


/* Change styles for span and cancel button on extra small screens */

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}
.cancelbtn {

width: 100%;

}
```

```
}

</style>

</head>

<body style="font-family:Montserrat;">

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Vehicle
Damage Detection</div>

<div class="topnav-right" >


<a href="{{ url_for('home')}}">Home</a>

<a href="{{ url_for('login')}}">Login</a>

<a class="active" href="{{ url_for('register')}}">Register</a>


</div>

</div>

<div id="login" class="login">


<form action="{{url_for('afterreg')}}" method="post">

<div class="imgcontainer">

<img style="" src="https://www.iconbunny.com/icons/media/catalog/product/2/1/2136.12-farmer-
icon-iconbunny.jpg" alt="Avatar" class="avatar">

</div>
```

```html
<div class="container">

<input type="text" placeholder="Enter Name" name="name" required><br>

<input type="email" placeholder="Enter Email ID" name="_id" required><br>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Register</button><br>

</div>

<div class="container" style="background-color:#f1f1f1">

<div class="psw">Already have an account?   <a href="{{ url_for('login') }}">Login</a></div >

</div>

</form>

</div>

</body>

</html>
```