PROJECT REPORT

Intelligent Vehicle Damage Assessment & CostEstimator For Insurance Companies Using IBMCloud

Submitted by:

PNT2022TMID45692

Team leader :G.B.POORNIMA
Team members:R.GOKILA
S.SELVADHARSHINI
A.BANUMATHI

INDEX

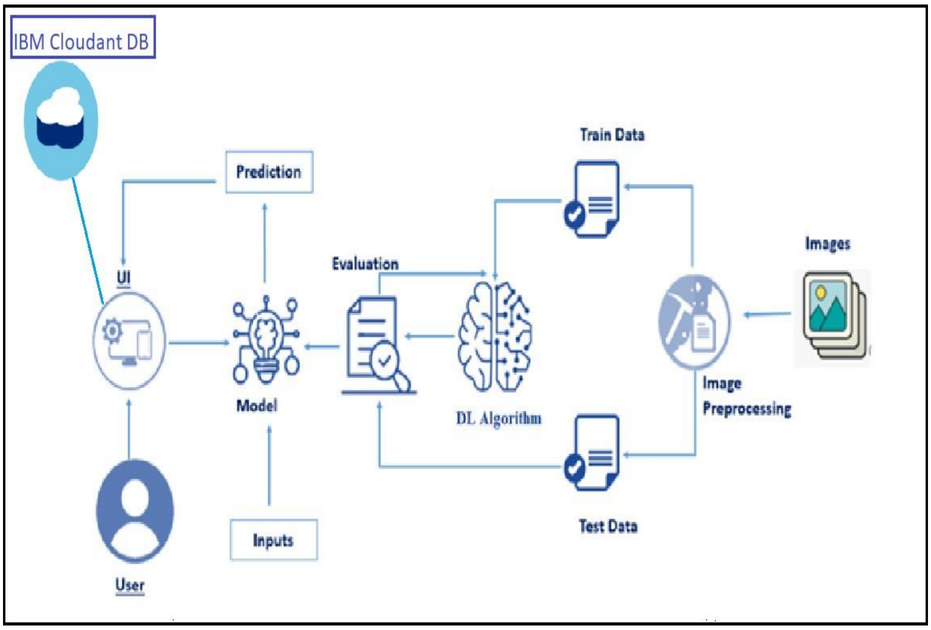
Intelligent Vehicle Damage Assessment & CostEstimator For Insurance Companies Using IBMCloud

INTRODUCTION:

1.1 Project 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 leadingpractices 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. Therationale 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 carloan especially for a used car.



1.2 Purpose:

This project aims at building a model, which is used to predict the estimated cost of the vechile damge.

The main purpose is to enchance customer experience and drive efficiency by reducing turnaround time for claim settlement.

Improving customer experience with smarter solutions.

Seeking to enhanced customer experience and drive efficiency,IFFCO Tokio collaborated with IBM Services to automated process using AI based solutions.

2.LIYERATURE SURVEY:

2.1 Existing problem:

With the popularity of the concept of sharing, car sharing has become a new hotspot. Currently, there are more than 100 brands operating car sharing in the market. The group composition of car rental is relatively complex, and most of them are novices in driving technology. Some minor scratches and other injuries are easy to occur during the driving of rental vehicles, with poses new challenges to the car-sharing operations platform.Therefore, in the operations of the car-sharing business, it is very important to realize the automatic determination of vehicles damage in the process of each use.

2.2 References :

- Zhu Qianqian 1,Guo Weiming1, Shen Ying2 and Zhao Zihoa,Journal of Physics:Conference Series.2022,Research on Intelligent Vehicle Damage Assessment System Based on Computer vision.
- Vikas Taliwal, Boston, MA (US); Siddartha Dalal, Bridgewater, NJ(US);Kaigang Li Brooklyn, NY(US).October 12,2017,Automatic assessment of damage and repair costs in vehicles.
- Aniket Gupta1, Jitesh Chogale2, Shashank Shrivastav3, Prof. Rupali Nikhare4, International Research Journal of Engineering and Technology (IRJET). April 2020, Automatic Car Insurance using Image Analysis.
- Girish N1, Mohammed Aqeel A rshad2, International Journal of Advanced Research in Science Communication and Technology (IJARSCT). 2020,Car Damage Detection using Machine Learning.
- Mohammed Yusuf Jamal Aziz Azmi Israr Ahmad,Mohammed ZainulArefeen, Daaniyal Ahmed, Hussam Bin Mehare,International Research Journal of Modernnization in Engineering Technology and Science 2022,vehicle Damage detection using deep learning.
- Mandara G S and Prashant Ankalkoti, International Journal of Advanced Research in Computer and Communication Engineering, 2020 Car Damage Assessment for Insurance Companies.
- Vaibhav Agarwal, UtsavKhandelwal,Shiva Kumar,Raja Kumar,Shilpa M,International Journal Of Creative Research Thoughts.2020,Damage Assessment Of a Vehicle and Insurance Reclaim.

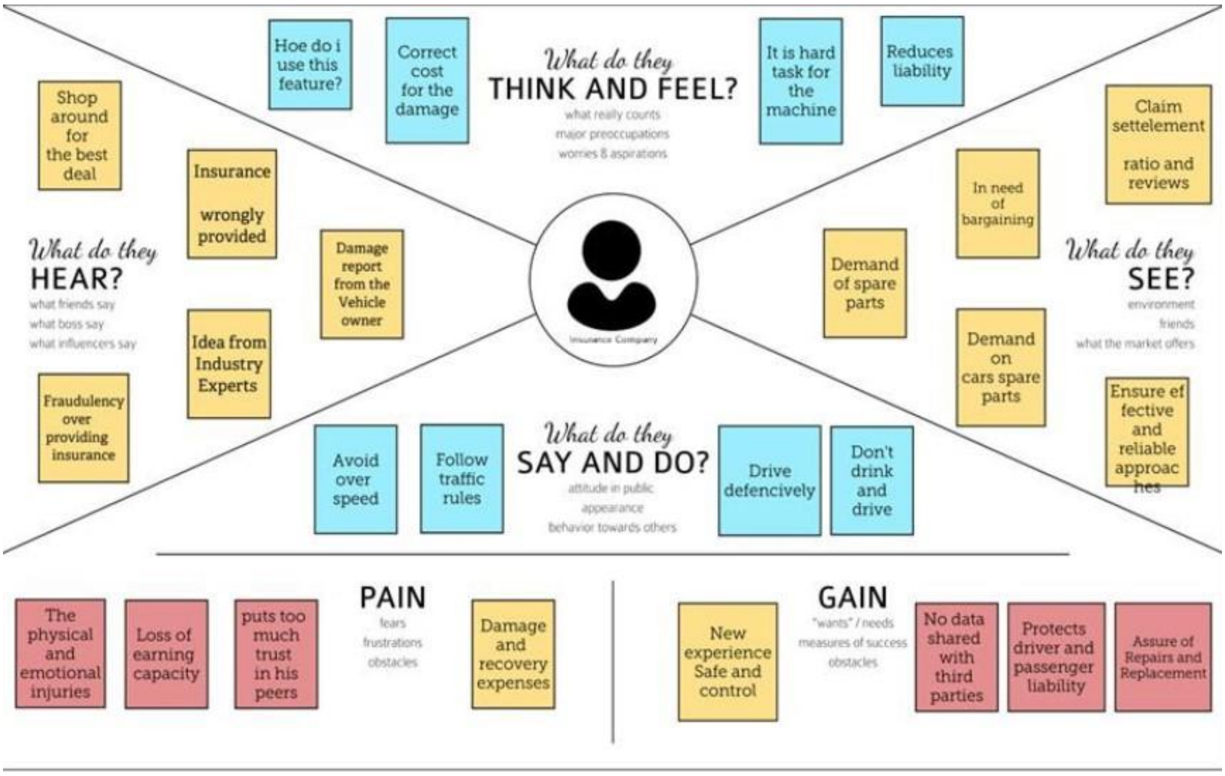
  Srimal Jayawardena,A Thesis Submitted For the degree of Doctor of Philosopy at The Australian National University.2021, Image Based Automatic Vehicle Damage Detection.
- Avinash Sharma, AaditiVerma, Dhananjay Gupta,International Journal of Innovation Technology and Exploring Engineering,November 2019,Preventing Car Dmage using CNN and Computer Vision.

2.3 Problem Satatement definition :

This paper presents a method and system for automatic damage determination of vehicles based on the shared vehicle four corner-images. By comparing the damage information before and after renting each vehicle,this scheme can effectively save labor cost, realize rapid damage recognition, clear ewsponsibility definition, and improve user expensive and damage treatment efficiency.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas:



Hoe do i use this feature?
Correct cost for the damage
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
It is hard task for the machine
Reduces liability
Shop around for the best deal
Insurance wrongly provided
Damage report from the Vehicle owner
What do they HEAR?
what friends say
what boss say
what influencers say
Idea from Industry Experts
Fraudulency over providing insurance
Insurance Company
Claim settelement ratio and reviews
In need of bargaining
Demand of spare parts
What do they SEE?
environment
friends
what the market offers
Demand on cars spare parts
Ensure ef fective and reliable approac hes
Avoid over speed
Follow traffic rules
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Drive defencively
Don't drink and drive
The physical and emotional injuries
Loss of earning capacity
puts too much trust in his peers
PAIN
fears
frustrations
obstacles
Damage and recovery expenses
New experience Safe and control
GAIN
"wants" / needs
measures of success
obstacles
No data shared with third parties
Protects driver and passenger liability
Assure of Repairs and Replacement

3.2 Ideation & Brainstroming:

PROBLEM STATEMENT:

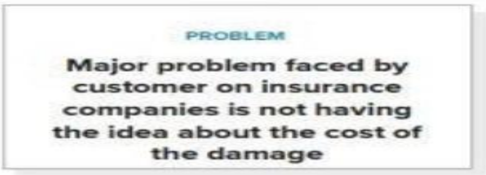
Major problem faced by the consumer on an insurance company is not having the idea about the cost of Damage.

Insurance companies are failing to provide the right amount for the car damage and the customer is not able to claim for the damage.

Developing the solution that can able to identify the right cost for the damage would be beneficial for many companies.

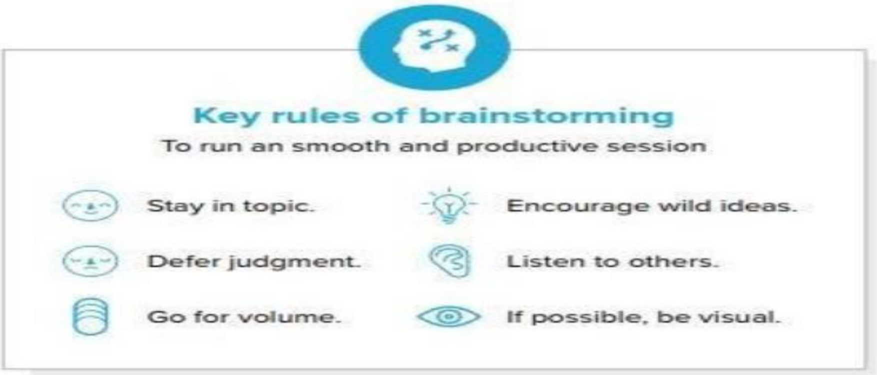
1. Define your problem statement

What problem are you trying to solve? Frame your problem as a How might we Statement. This will be focus on your BRAINSTROM.



PROBLEM

Major problem faced by customer on insurance companies is not having the idea about the cost of the damage



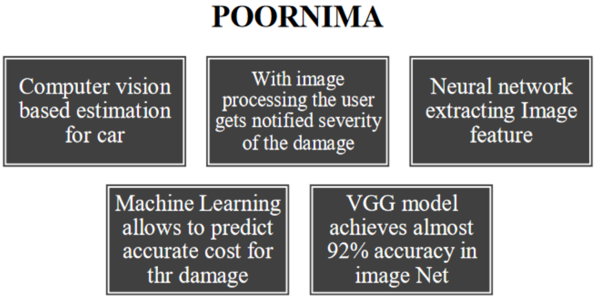
Key rules of brainstorming

To run an smooth and productive session

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

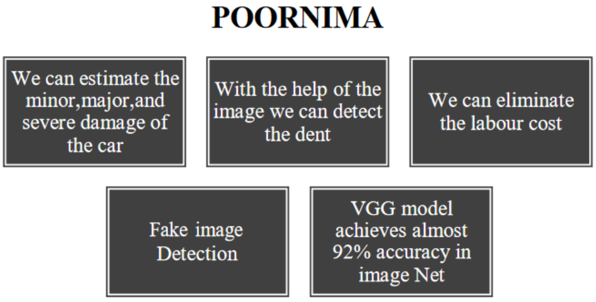
2. Brainstorm

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



POORNIMA

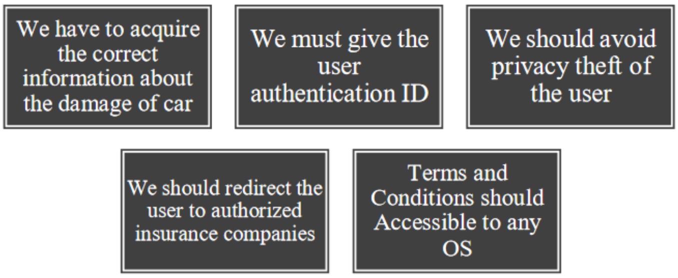
- Computer vision based estimation for car
- With image processing the user gets notified severity of the damage
- Neural network extracting Image feature
- Machine Learning allows to predict accurate cost for thr damage
- VGG model achieves almost 92% accuracy in image Net



POORNIMA

- We can estimate the minor,major,and severe damage of the car
- With the help of the image we can detect the dent
- We can eliminate the labour cost
- Fake image Detection
- VGG model achieves almost 92% accuracy in image Net

GOKILA


We have to acquire the correct information about the damage of car

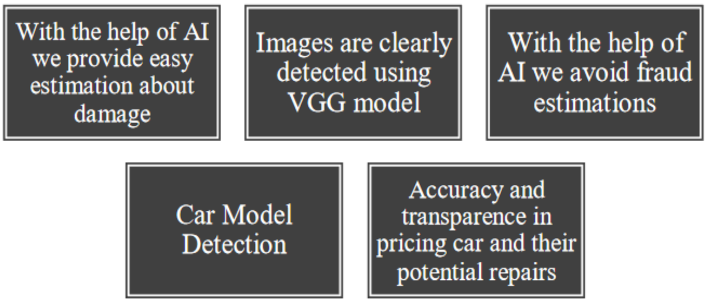
We must give the user authentication ID

We should avoid privacy theft of the user

We should redirect the user to authorized insurance companies

Terms and Conditions should Accessible to any OS

BANUMATHI


With the help of AI we provide easy estimation about damage

Images are clearly detected using VGG model

With the help of AI we avoid fraud estimations

Car Model Detection

Accuracy and transparence in pricing car and their potential repairs

3.3 Proposed Solution :

Insurance companyfrequently suffer loses.Because they did notprovide a properexplanation regards the estimation of the damage to the customer.

We create an Ai Model to sense and detect the precise amount damage that occurred in the vehicle.
Then we create a user accessible portal and securely store the data provided by the user.
Finally compare the gathered damage percentage with the statistical cost estimation value to predict the cost.

The AI Model automatically calculates the damaged vehicle cost.
The deep learning algorithm provides progressively higher level features

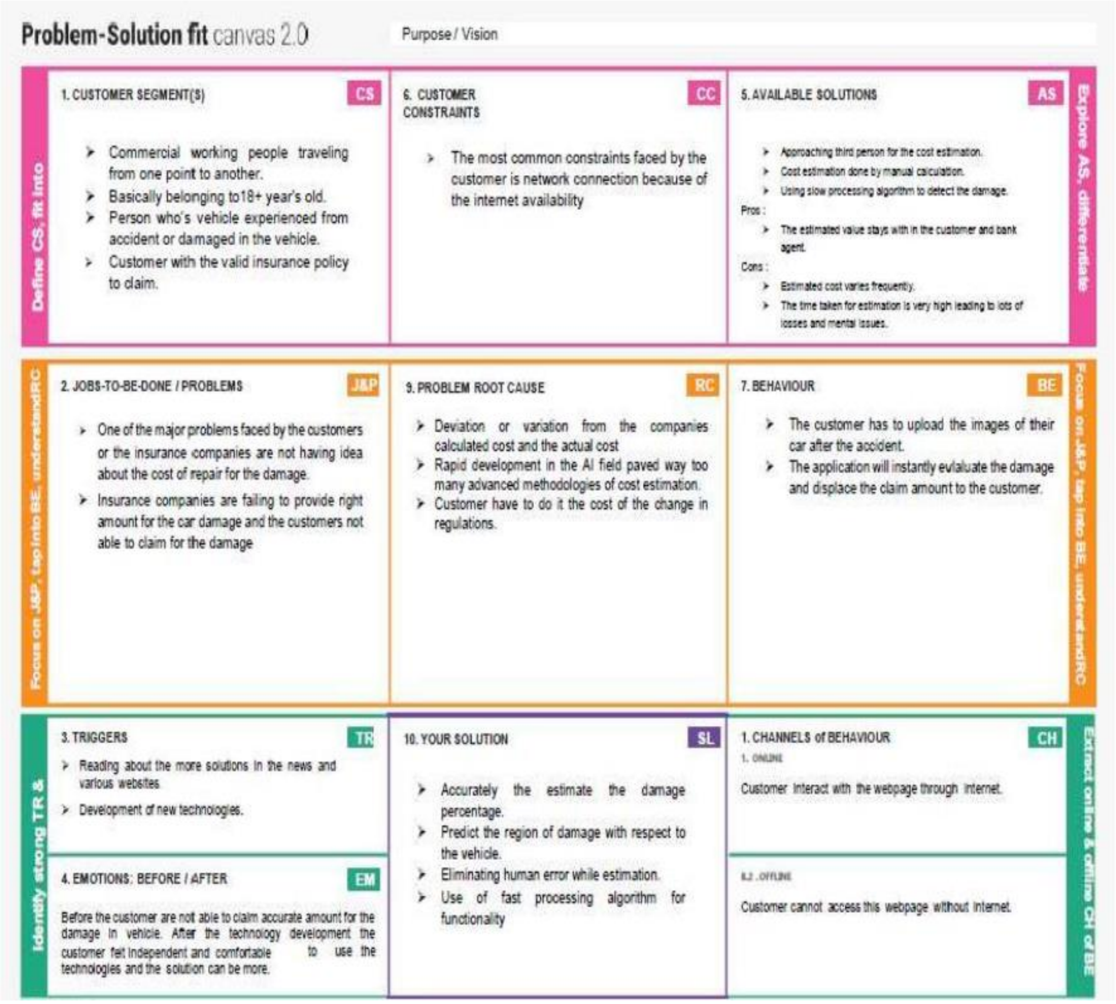
It's the user friendly website.

All the images and personal data will be secured in the cloud data security

Insurance companies have two primary sources of income Underwriting & Investment income.
Financial investments including Listed shares, Government bonds, and Corporate bonds, make up the majority of insurance firms' assets.
By estimating the level of car damage using our AI model and providing insurance accordingly, they can save more money and invest it in their businesses.

With the use of advanced machine learning techniques analyze damaged vehicles with high accuracy levels and keep on improving the learning ability of the model.
Our AI model can operate at the scale, speed, and complexity required for the aim.

3.4 Problem solution fit:



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into

1. CUSTOMER SEGMENT(S) CS

- Commercial working people traveling from one point to another.
- Basically belonging to18+ year's old.
- Person who's vehicle experienced from accident or damaged in the vehicle.
- Customer with the valid insurance policy to claim.

6. CUSTOMER CONSTRAINTS CC

- The most common constraints faced by the customer is network connection because of the internet availability

5. AVAILABLE SOLUTIONS AS

- Approaching third person for the cost estimation.
- Cost estimation done by manual calculation.
- Using slow processing algorithm to detect the damage.

Pros :

- The estimated value stays with in the customer and bank agent.

Cons :

- Estimated cost varies frequently.
- The time taken for estimation is very high leading to lots of losses and mental issues.

Explore AS, differentiate

Focus on J&P, tap into BE, understandRC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- One of the major problems faced by the customers or the insurance companies are not having idea about the cost of repair for the damage.
- Insurance companies are failing to provide right amount for the car damage and the customers not able to claim for the damage

9. PROBLEM ROOT CAUSE RC

- Deviation or variation from the companies calculated cost and the actual cost
- Rapid development in the AI field paved way too many advanced methodologies of cost estimation.
- Customer have to do it the cost of the change in regulations.

7. BEHAVIOUR BE

- The customer has to upload the images of their car after the accident.
- The application will instantly evlaluate the damage and displace the claim amount to the customer.

Focus on J&P, tap into BE, understandRC

Identify strong TR & EM

3. TRIGGERS TR

- Reading about the more solutions in the news and various websites
- Development of new technologies.

4. EMOTIONS: BEFORE / AFTER EM

Before the customer are not able to claim accurate amount for the damage in vehicle. After the technology development the customer felt independent and comfortable to use the technologies and the solution can be more.

10. YOUR SOLUTION SL

- Accurately the estimate the damage percentage.
- Predict the region of damage with respect to the vehicle.
- Eliminating human error while estimation.
- Use of fast processing algorithm for functionality

1. CHANNELS of BEHAVIOUR CH

1. ONLINE

Customer interact with the webpage through internet.

8.2 .OFFLINE

Customer cannot access this webpage without internet.

Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS:

4.1 Funtional requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registeration	Download the app Registration through Gmail Create an account Follow the instructions
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP

FR-3	Interface	Good Interface to user to operate
FR-4	Accessing datasets	Details about user Details about vehicle Details about insurance companies
FR-5	Mobile application	AI and camera sensor in the field can be access by mobile application.

4.2 Non functional requirements:

Following are the non-functional requirements of the proposed solution.

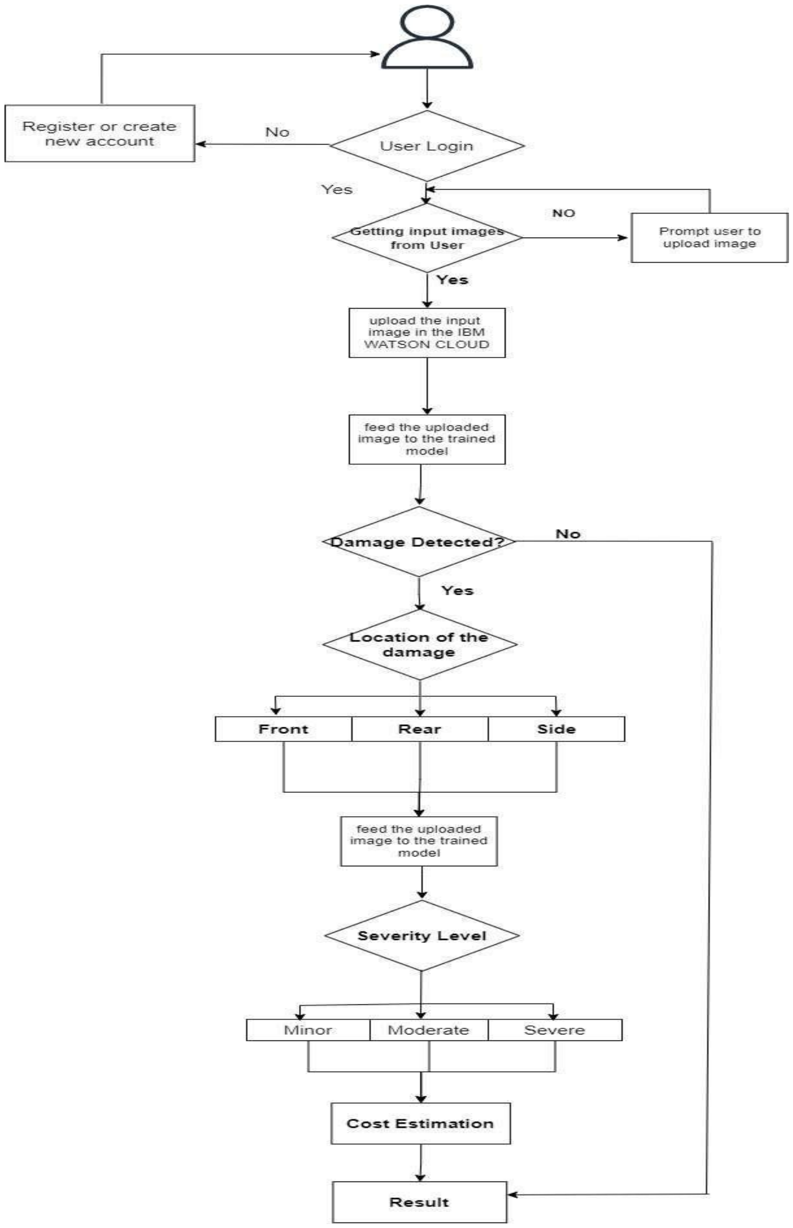
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The smart claiming system for vehicle damage insurance in bank companies
NFR-2	**Security**	We have designed this project to user easy to claim the insurance .
NFR-3	**Reliability**	This project will help the user to claim the insurance cost based on vehicle damage. It gives the exact value to user. This helps user to get correct cost without any failure.
NFR-4	**Performance**	AI devices and sensors are used to indicate the user to estimated the cost of the vehicle.AI camera to scan the damaged vehicle and gives exact cost insurance to user.
NFR-5	**Availability**	This application is designed for all devices and also vailable in apk.
NFR-6	**Scalability**	This project is more scalability in our present and future uses to estimate the cost exactly to user.

5.PROJECT DESIGN:

5.1 Data Flow Diagrams:

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored


Register or create new account
No
User Login
Yes
NO
Getting input images from User
Prompt user to upload image
Yes
upload the input image in the IBM WATSON CLOUD
feed the uploaded image to the trained model
Damage Detected?
No
Yes
Location of the damage
Front
Rear
Side
feed the uploaded image to the trained model
Severity Level
Minor
Moderate
Severe
Cost Estimation
Result

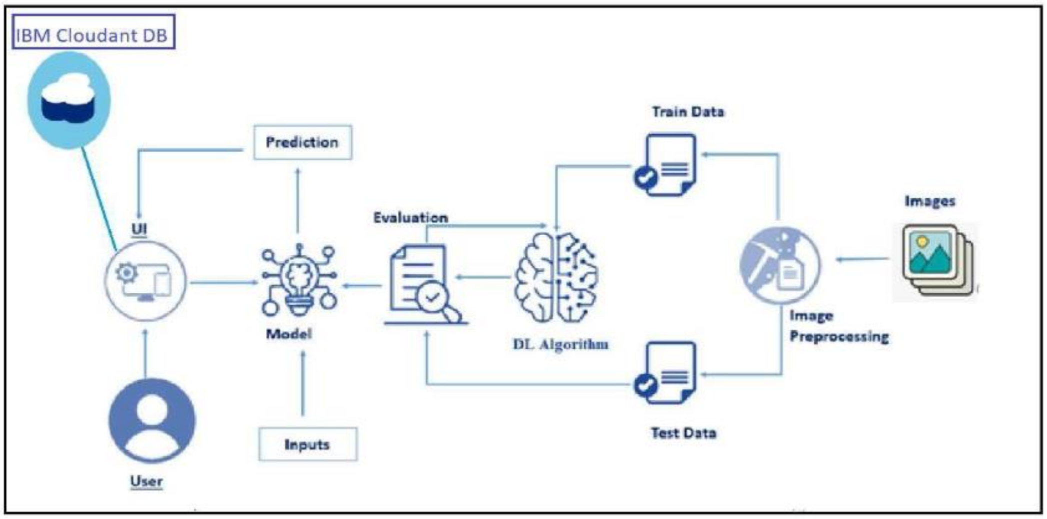
5.2 Solution & Technical architecture:

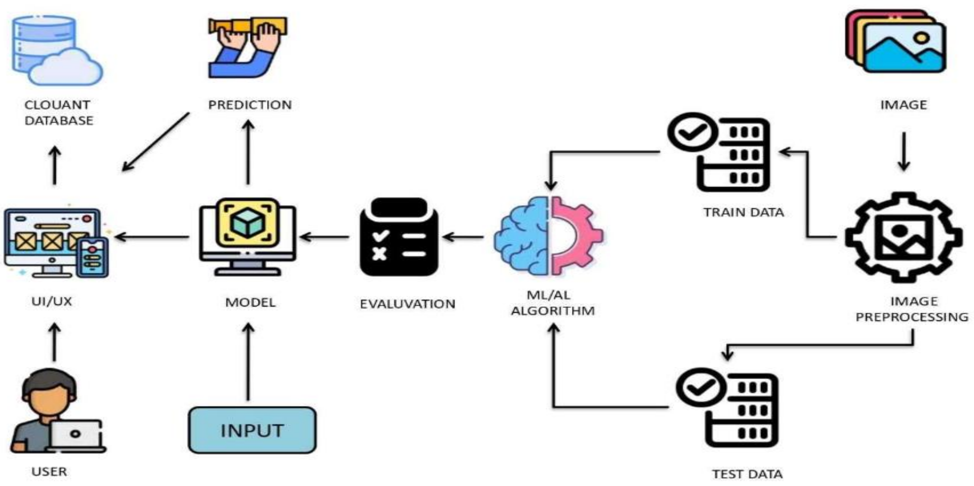
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

Find the best tech solution to solve existing business problems.

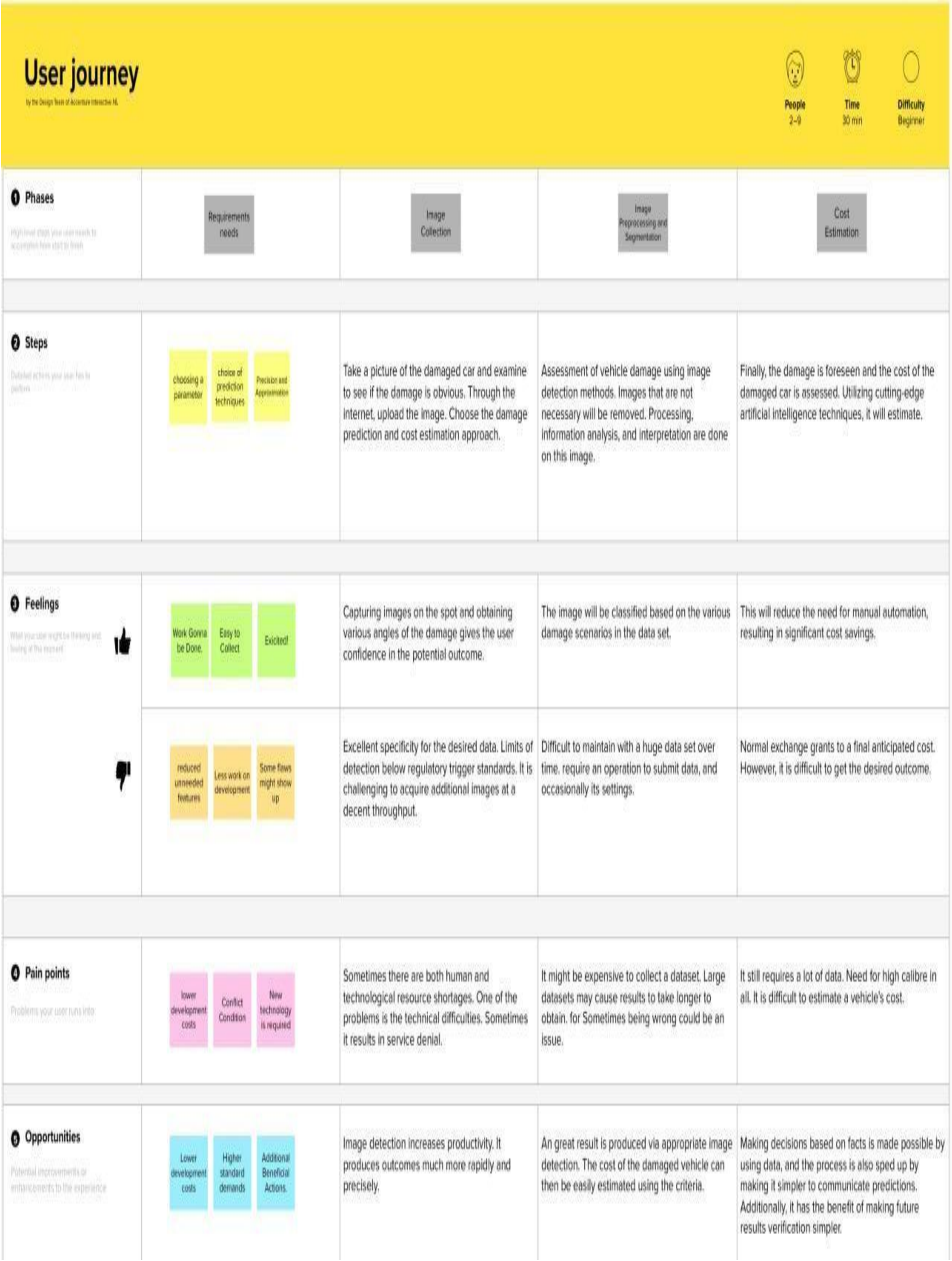
Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.





5.3 User stories:



User journey

by the Design Team of Accenture Interactive NL

People 2–9

Time 30 min

Difficulty Beginner

❶ **Phases** High level steps your user needs to accomplish from start to finish	Requirements needs	Image Collection	Image Preprocessing and Segmentation	Cost Estimation
❷ **Steps** Detailed actions your user has to perform	choosing a parameter choice of prediction techniques Precision and Approximation	Take a picture of the damaged car and examine to see if the damage is obvious. Through the internet, upload the image. Choose the damage prediction and cost estimation approach.	Assessment of vehicle damage using image detection methods. Images that are not necessary will be removed. Processing, information analysis, and interpretation are done on this image.	Finally, the damage is foreseen and the cost of the damaged car is assessed. Utilizing cutting-edge artificial intelligence techniques, it will estimate.
❸ **Feelings** What your user might be thinking and feeling at the moment 👍	Work Gonna be Done. Easy to Collect Excited!	Capturing images on the spot and obtaining various angles of the damage gives the user confidence in the potential outcome.	The image will be classified based on the various damage scenarios in the data set.	This will reduce the need for manual automation, resulting in significant cost savings.
👎	reduced unneeded features Less work on development Some flaws might show up	Excellent specificity for the desired data. Limits of detection below regulatory trigger standards. It is challenging to acquire additional images at a decent throughput.	Difficult to maintain with a huge data set over time. require an operation to submit data, and occasionally its settings.	Normal exchange grants to a final anticipated cost. However, it is difficult to get the desired outcome.
❹ **Pain points** Problems your user runs into	lower development costs Conflict Condition New technology is required	Sometimes there are both human and technological resource shortages. One of the problems is the technical difficulties. Sometimes it results in service denial.	It might be expensive to collect a dataset. Large datasets may cause results to take longer to obtain. for Sometimes being wrong could be an issue.	It still requires a lot of data. Need for high calibre in all. It is difficult to estimate a vehicle's cost.
❺ **Opportunities** Potential improvements or enhancements to the experience	Lower development costs Higher standard demands Additional Beneficial Actions.	Image detection increases productivity. It produces outcomes much more rapidly and precisely.	An great result is produced via appropriate image detection. The cost of the damaged vehicle can then be easily estimated using the criteria.	Making decisions based on facts is made possible by using data, and the process is also sped up by making it simpler to communicate predictions. Additionally, it has the benefit of making future results verification simpler.

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	Registration	USN - 1	As a user, I can register for the application by entering my details of name, email, cars etc. verifying my Gmail account and creating new account with password	7	HIGH	TM-1,4
Sprint-1	Login	USN -2	As a user,entering my email, and password, and confirming my password,I can login to my account.	7	HIGH	TM-1,4
Sprint-1	Dashboard	USN-3	As a user, I can clearly see data, point, graphs, charts and trends of my previous activity and global activity related to my views	2	LOW	TM-1,4
Sprint-2	Details about insurance company	USN-4	As a user, I can register for theApplication through Gmail and account id.	8	MEDIUM	`TM-2,3
Sprint-1	repeated logins and logout	USN-5	As a user,I can log in and view my dashboard at my demand on any time	4	HIGH	TM-1,4
Sprint-2	Webpage	USN-6	As a user, I must enter all details of car, accident, capture images of my vehicle and upload it into the web portal.	12	HIGH	TM-2,3
Sprint 3	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the Cost estimated.	20	HIGH	TM-1,2
Sprint 4	Provide friendly and efficient	USN-8	As a user, I need to get support from developers in case of	10	MEDIUM	TM-1,2,3

	customer support and sort out the queries.		queries and failure of service Provided by chat-box,mail or call.			
Sprint 4	overview the entire process and act as a bridge between user and developer	USN-9	As a Team member, We need to satisfy thecustomer needs in an efficient way and make sure any sort of errors are fixed	10	HIGH	TM-1,2,3

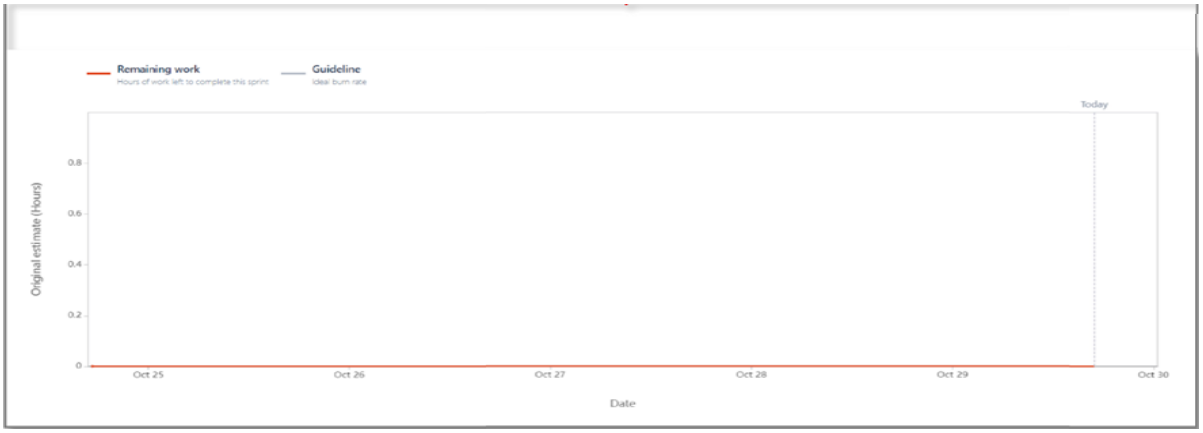
6.2 Sprint Delivery Schedule:

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

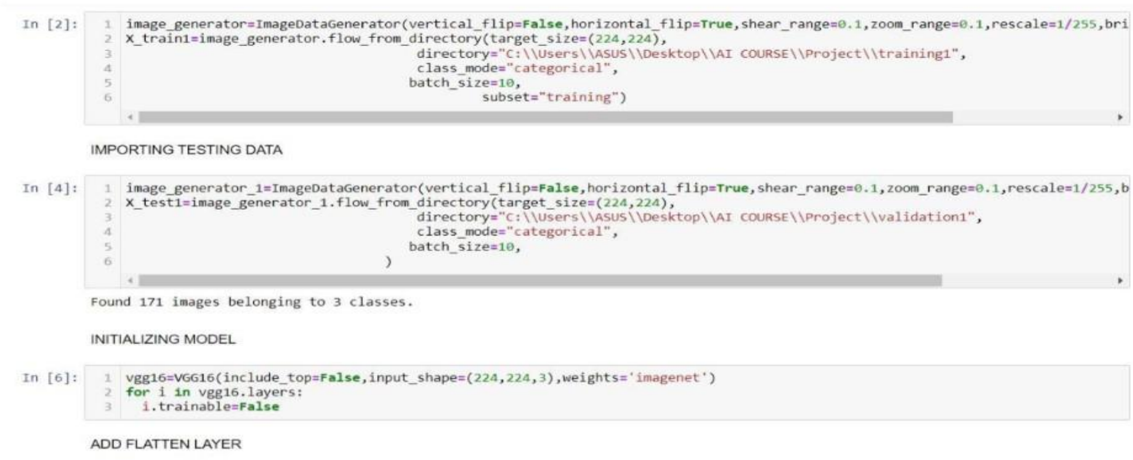
6.3 Reports from JIRA:

$$AV = \frac{SPRINT\ DURATION}{VELOCITY} = \frac{20}{6} = 3.33$$

Burn-down Chart:



7. CODING & SOLUTIONING (explain the features added in the project along with code)



```
In [2]:  image_generator=ImageDataGenerator(vertical_flip=False,horizontal_flip=True,shear_range=0.1,zoom_range=0.1,rescale=1/255,bri
         X_train1=image_generator.flow_from_directory(target_size=(224,224),
                                                  directory="C:\\Users\\ASUS\\Desktop\\AI COURSE\\Project\\training1",
                                                  class_mode="categorical",
                                                 batch_size=10,
                                                          subset="training")
```

IMPORTING TESTING DATA

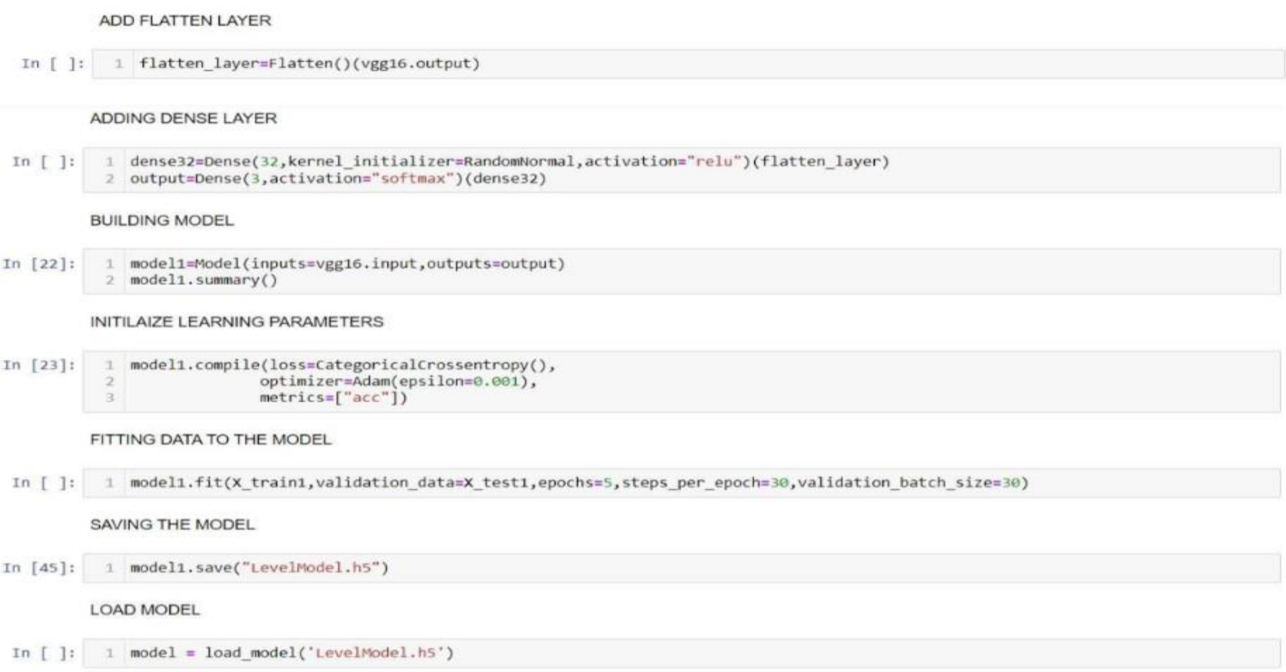
```
In [4]:  image_generator_1=ImageDataGenerator(vertical_flip=False,horizontal_flip=True,shear_range=0.1,zoom_range=0.1,rescale=1/255,b
         X_test1=image_generator_1.flow_from_directory(target_size=(224,224),
                                                  directory="C:\\Users\\ASUS\\Desktop\\AI COURSE\\Project\\validation1",
                                                  class_mode="categorical",
                                                 batch_size=10,
                                              )
```

```
Found 171 images belonging to 3 classes.
```

INITIALIZING MODEL

```
In [6]:  vgg16=VGG16(include_top=False,input_shape=(224,224,3),weights='imagenet')
         for i in vgg16.layers:
           i.trainable=False
```

ADD FLATTEN LAYER



ADD FLATTEN LAYER

```
In [ ]:  flatten_layer=Flatten()(vgg16.output)
```

ADDING DENSE LAYER

```
In [ ]:  dense32=Dense(32,kernel_initializer=RandomNormal,activation="relu")(flatten_layer)
         output=Dense(3,activation="softmax")(dense32)
```

BUILDING MODEL

```
In [22]:  model1=Model(inputs=vgg16.input,outputs=output)
          model1.summary()
```

INITILAIZE LEARNING PARAMETERS

```
In [23]:  model1.compile(loss=CategoricalCrossentropy(),
                         optimizer=Adam(epsilon=0.001),
                         metrics=["acc"])
```

FITTING DATA TO THE MODEL

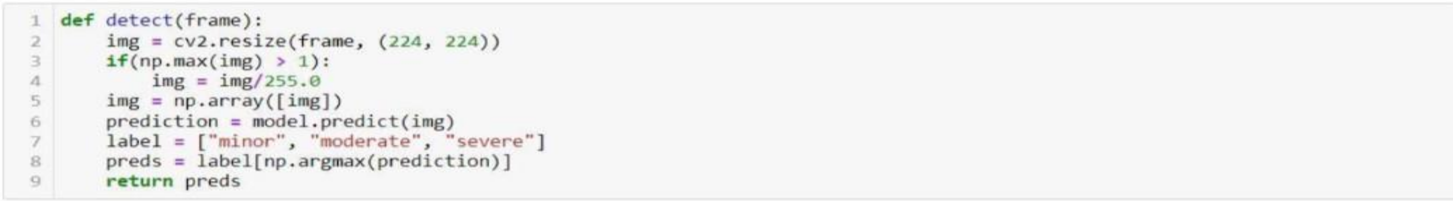
```
In [ ]:  model1.fit(X_train1,validation_data=X_test1,epochs=5,steps_per_epoch=30,validation_batch_size=30)
```

SAVING THE MODEL

```
In [45]:  model1.save("LevelModel.h5")
```

LOAD MODEL

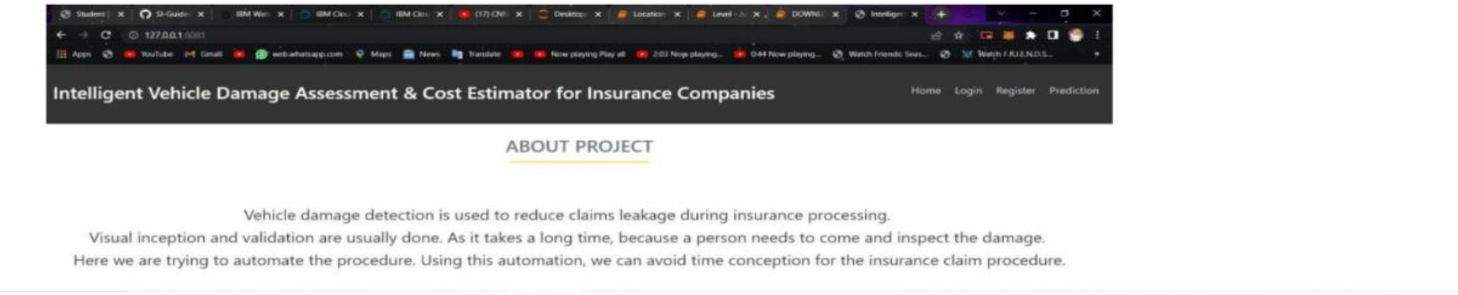
```
In [ ]:  model = load_model('LevelModel.h5')
```



```
def detect(frame):
    img = cv2.resize(frame, (224, 224))
    if(np.max(img) > 1):
        img = img/255.0
    img = np.array([img])
    prediction = model.predict(img)
    label = ["minor", "moderate", "severe"]
    preds = label[np.argmax(prediction)]
    return preds
```

Output:

Flask application user interface



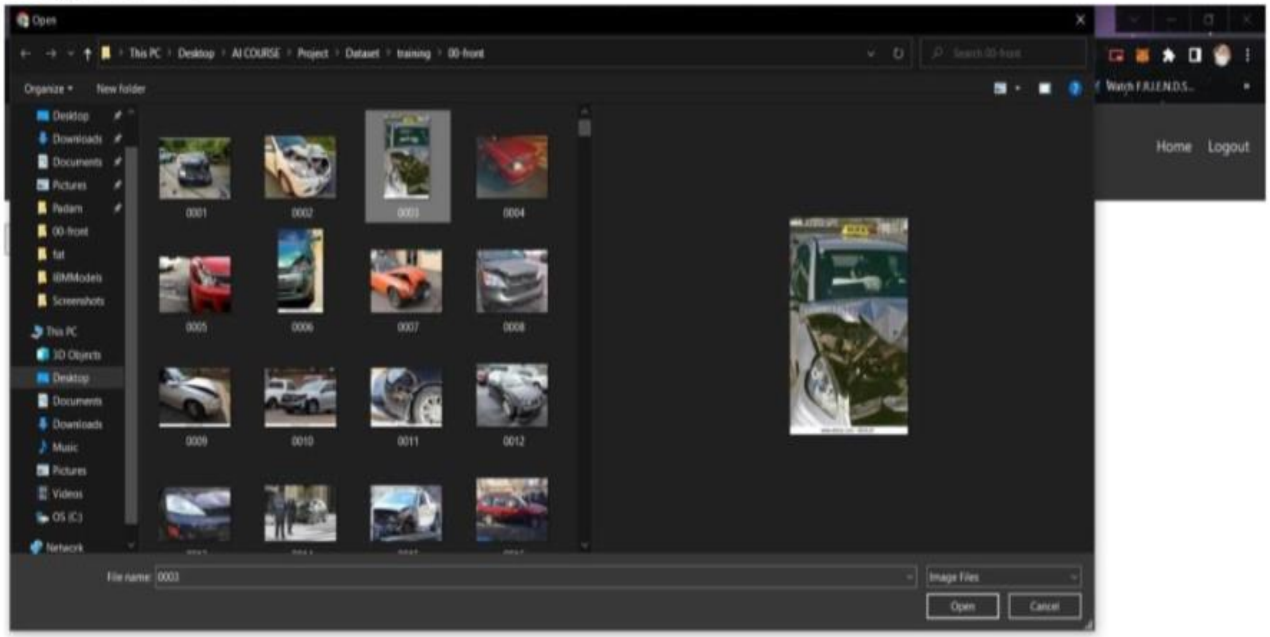
Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies

ABOUT PROJECT

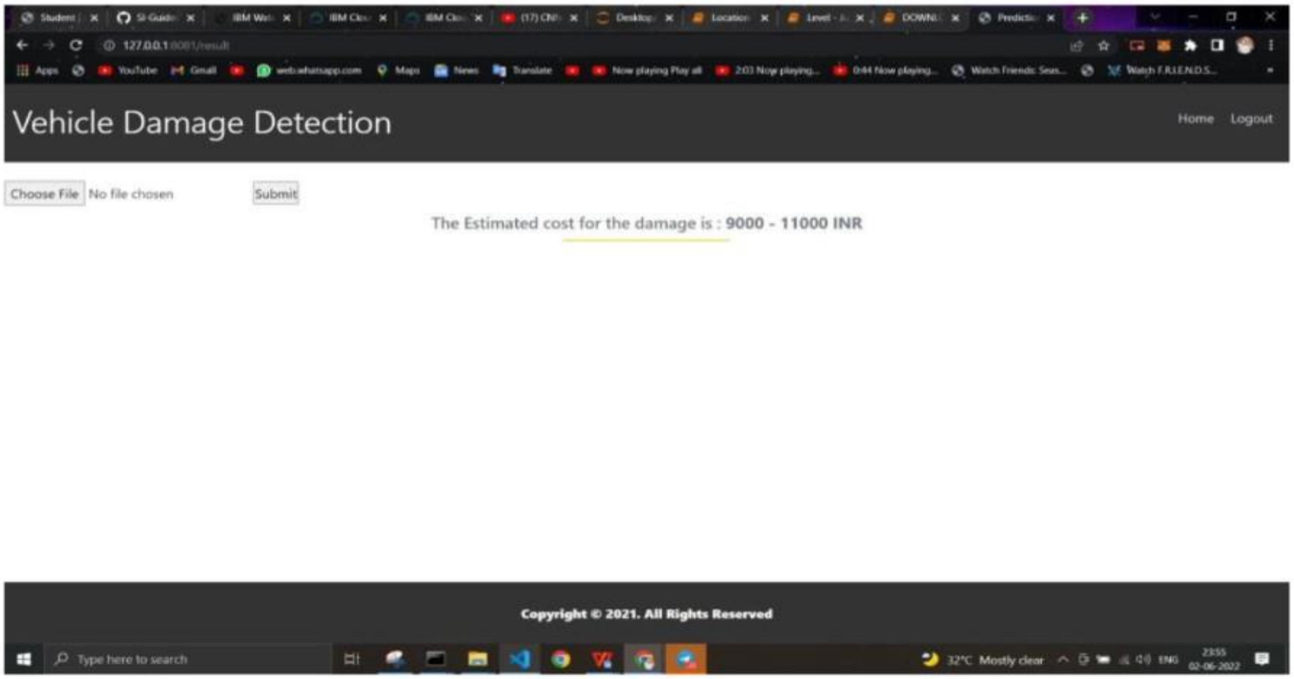
Vehicle damage detection is used to reduce claims leakage during insurance processing.
Visual inception and validation are usually done. As it takes a long time, because a person needs to come and inspect the damage.
Here we are trying to automate the procedure. Using this automation, we can avoid time conception for the insurance claim procedure.

Copyright © 2021. All Rights Reserved

Image choosed



File name: 0003



The Estimated cost for the damage is : 9000 - 11000 INR

Copyright © 2021. All Rights Reserved

INPUT => Input given is Front and severely damaged vehicle

OUTPUT=> Output got is “front severe”.

BUILD PYTHON CODE:

```
  import os
  import   h5py import
  numpy as npimport json
  import urllib.request
  import numpy as np
  import matplotlib.pyplot as plt
  import pandas as pd
  import seaborn as sns
  from IPython.display import Image, display, clear_output
from sklearn.metrics import classification_report, confusion_matrix

  %matplotlib inline
  sns.set_style('whitegrid')
                                                                In [2]:
  from keras import optimizers
  from keras.applications.vgg16 import VGG16
  from keras.models import Sequential, load_model, Model
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D, Activation,
Dropout, Flatten, Dense, Input
```

```
  from keras.regularizers import l2, l1
  from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator, array_to_img,img_to_array,
load_img
  from keras.callbacks import ModelCheckpoint, History
  from keras import backend as K
from keras.utils.data_utils import get_fileUsing
TensorFlow backend.
```

In [3]:

```
def plot_metrics(hist, stop=50):
        fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10,4))axes =

        axes.flatten()

      axes[0].plot(range(stop), hist['acc'], label='Training',color='#FF533D')
      axes[0].plot(range(stop), hist['val_acc'], label='Validation',color='#03507E')
        axes[0].set_title('Accuracy')
        axes[0].set_ylabel('Accuracy')
        axes[0].set_xlabel('Epoch')
        axes[0].legend(loc='lower right'
        axes[1].plot(range(stop), hist['loss'], label='Training',color='#FF533D')
      axes[1].plot(range(stop), hist['val_loss'], label='Validation',color='#03507E')
        axes[1].set_title('Loss')
        axes[1].set_ylabel('Loss')
        axes[1].set_xlabel('Epoch')
        axes[1].legend(loc='upper right')

        plt.tight_layout();

        print("Best Model:")
        print_best_model_results(hist)
```

In [4]:

```
def plot_acc_metrics(hist1, hist2, stop=50):
    fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(4.25,6))axes =

    axes.flatten()

    axes[0].plot(range(stop), hist1['acc'], label='Training',color='#FF533D')
    axes[0].plot(range(stop), hist1['val_acc'], label='Validation',color='#03507E')
    axes[0].set_title('Training')
    axes[0].set_ylabel('Accuracy')
    axes[0].set_xlabel('Epoch')
    axes[0].legend(loc='lower right')

    axes[1].plot(range(stop), hist2['acc'], label='Training',color='#FF533D')
    axes[1].plot(range(stop), hist2['val_acc'], label='Validation',color='#03507E')
    axes[1].set_title('Fine-tuning')
    axes[1].set_ylabel('Accuracy')
    axes[1].set_xlabel('Epoch')
    axes[1].legend(loc='lower right')

    plt.tight_layout();
```

In [5]:

```
def print_best_model_results(model_hist): best_epoch =
    np.argmax(model_hist['val_acc'])print('epoch:',
    best_epoch+1, \
    ', val_acc:', model_hist['val_acc'][best_epoch], \', val_loss:',
    model_hist['val_loss'][best_epoch])
```

In [6]:

```
def save_bottleneck_features():
    datagen = ImageDataGenerator(rescale=1./255)

    model = VGG16(include_top=False, weights='imagenet') generator =

    datagen.flow_from_directory(train_data_dir,
target_size=(img_width, img_height), batch_size=batch_size,
class_mode=None, shuffle=False)
   bottleneck_features_train = model.predict_generator(generator,nb_train_samples //
batch_size)
   np.save(location+'/bottleneck_features_train.npy',
bottleneck_features_train)

   generator = datagen.flow_from_directory(validation_data_dir,
target_size=(img_width, img_height), batch_size=batch_size, class_mode=None,
shuffle=False)
   bottleneck_features_validation = model.predict_generator(generator,
nb_validation_samples // batch_size)
   np.save(location+'/bottleneck_features_validation.npy',bottleneck_features_validation)
```

In [7]:

```
def train_top_model():
    train_data = np.load(location+'/bottleneck_features_train.npy')train_labels =
    np.array([0] * (nb_train_samples // 2) + [1] *
(nb_train_samples // 2))

    validation_data =
 np.load(location+'/bottleneck_features_validation.npy') validation_labels = np.array([0] *
    (nb_validation_samples // 2) + [1]
* (nb_validation_samples // 2))

    model = Sequential() model.add(Flatten(input_shape=train_data.shape[1:]))
    model.add(Dense(256,activation='relu')) model.add(Dropout(0.5))
    model.add(Dense(1,activation='sigmoid')) model.compile(optimizer='rmsprop',
    loss='binary_crossentropy',
metrics=['accuracy'])

    checkpoint = ModelCheckpoint(top_model_weights_path,
monitor='val_acc', verbose=1, save_best_only=True, save_weights_only=True,mode='auto')

    fit = model.fit(train_data, train_labels, epochs=epochs,
batch_size=batch_size,validation_data=(validation_data,validation_labels),callbacks=[checkpoint])

     with open(location+'/top_history.txt', 'w') as f:
         json.dump(fit.history, f)

     return model, fit.history
```

In [8]:

```
def finetune_binary_model():
    base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(256,256,3))
     print("Model loaded.")

     top_model = Sequential()
     top_model.add(Flatten(input_shape=base_model.output_shape[1:]))
     top_model.add(Dense(256, activation='relu')) top_model.add(Dropout(0.5))
     top_model.add(Dense(1, activation='sigmoid'))

     top_model.load_weights(top_model_weights_path)

    model = Model(inputs=base_model.input,
outputs=top_model(base_model.output))

     for layer in model.layers[:25]:layer.trainable
         = False

    model.compile(loss='binary_crossentropy', optimizer=optimizers.SGD(lr=1e-4,
momentum=0.9), metrics=['accuracy'])

    train_datagen = ImageDataGenerator(rescale = 1./255, zoom_range=0.2,shear_range=0.2,
horizontal_flip=True)

     test_datagen = ImageDataGenerator(rescale=1./255)

    train_generator = train_datagen.flow_from_directory(train_data_dir,
target_size=(img_height, img_width), batch_size=batch_size, class_mode='binary')
```

```
    validation_generator =
test_datagen.flow_from_directory(validation_data_dir,
target_size=(img_height, img_width), batch_size=batch_size,
class_mode='binary')

    checkpoint = ModelCheckpoint(fine_tuned_model_path, monitor='val_acc',verbose=1,
save_best_only=True, save_weights_only=False, mode='auto')

    fit = model.fit_generator(train_generator,
steps_per_epoch=nb_train_samples//batch_size, epochs=epochs,
validation_data=validation_generator,
validation_steps=nb_validation_samples//batch_size, verbose=1,
callbacks=[checkpoint])

    with open(location+'/ft_history.txt', 'w') as f:
        json.dump(fit.history, f)

    return model, fit.history
```

In [9]:

```
def evaluate_binary_model(model, directory, labels):datagen =
    ImageDataGenerator(rescale=1./255)

    generator = datagen.flow_from_directory(directory,
target_size=(img_height,img_width), batch_size=batch_size,
class_mode='binary', shuffle=False)

    predictions = model.predict_generator(generator, len(labels))

    pred_labels = [0 if i<0.5 else 1 for i in predictions]print('')
    print(classification_report(validation_labels, pred_labels))print('')
    cm = confusion_matrix(validation_labels, pred_labels)
    return cm
```

Defining input data

In [10]:

```
location = 'data2'
top_model_weights_path = location+'/top_model_weights.h5'
fine_tuned_model_path = location+'/ft_model.h5'

train_data_dir = location+'/training'
```

```
validation_data_dir = location+'/validation'
train_samples = [len(os.listdir(train_data_dir+'/'+i)) for i in
sorted(os.listdir(train_data_dir))]
nb_train_samples = 1824
validation_samples = [len(os.listdir(validation_data_dir+'/'+i)) for i in
sorted(os.listdir(validation_data_dir))]
nb_validation_samples = 448

img_width, img_height = 256,256epochs
= 50
batch_size = 16
```

In [ ]:

```
save_bottleneck_features()
```

In [11]:

```
d2_model1, d2_history1 = train_top_model() WARNING:tensorflow:From
C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from
tensorflow.python.framework.ops) is deprecated and will be removed ina future version.
Instructions for updating:
Colocations handled automatically by placer. WARNING:tensorflow:From
C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future
version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1
- keep_prob`.
WARNING:tensorflow:From C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in afuture version.
Instructions for updating:
Use tf.cast instead.
Train on 1824 samples, validate on 448 samplesEpoch 1/50
1824/1824 [==============================] - 19s 10ms/step - loss: 7.9614
- acc: 0.5016 - val_loss: 8.0590 - val_acc: 0.5000

Epoch 00001: val_acc improved from -inf to 0.50000, saving model to
data2/top_model_weights.h5
Epoch 2/50
1824/1824 [==============================] - 19s 10ms/step - loss: 8.0590
```

```
   - acc: 0.5000 - val_loss: 8.0590 - val_acc: 0.5000

   Epoch 00002: val_acc did not improve from 0.50000Epoch 3/50
   1824/1824 [==============================] - 18s 10ms/step
                                                                        - loss:    8.0590
   - acc: 0.5000 - val_loss: 8.0590 - val_acc: 0.5000

   Epoch 00003: val_acc did not improve from 0.50000Epoch 4/50
   1824/1824 [==============================] - 18s 10ms/step
                                                                        - loss:    8.0590
   - acc: 0.5000 - val_loss: 8.0590 - val_acc: 0.5000

   Epoch 00004: val_acc did not improve from 0.50000Epoch 5/50
   1824/1824 [==============================] - 20s 11ms/step
                                                                        - loss:    4.0526
- acc: 0.6552 - val_loss: 0.5081 - val_acc: 0.8036

Epoch 00005: val_acc improved from 0.50000 to 0.80357, saving model to
data2/top_model_weights.h5
Epoch 6/50
1824/1824 [==============================] - 19s 11ms/step - loss: 0.7258
- acc: 0.8026 - val_loss: 0.4214 - val_acc: 0.8549

Epoch 00006: val_acc improved from 0.80357 to 0.85491, saving model to
data2/top_model_weights.h5
Epoch 7/50
1824/1824 [==============================] - 19s 10ms/step - loss: 0.4354
- acc: 0.8520 - val_loss: 0.2513 - val_acc: 0.9174

Epoch 00007: val_acc improved from 0.85491 to 0.91741, saving model to
data2/top_model_weights.h5
Epoch 8/50
1824/1824 [==============================] - 20s 11ms/step - loss: 0.4085
- acc: 0.8739 - val_loss: 0.5095 - val_acc: 0.8460

Epoch 00008: val_acc did not improve from 0.91741Epoch 9/50
1824/1824 [==============================] - 20s 11ms/step - loss: 0.2958
- acc: 0.8964 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00009: val_acc improved from 0.91741 to 0.93750, saving model to
data2/top_model_weights.h5
Epoch 10/50
```

```
1824/1824 [==============================] - 21s 12ms/step                    - loss:    0.2781
- acc: 0.9052 - val_loss: 0.2311 - val_acc: 0.9286

Epoch 00010: val_acc did not improve from 0.93750Epoch 11/50
1824/1824 [==============================] - 21s 12ms/step
                                                                             - loss:    0.2184
- acc: 0.9216 - val_loss: 0.2545 - val_acc: 0.9286

Epoch 00011: val_acc did not improve from 0.93750Epoch 12/50
1824/1824 [==============================] - 20s 11ms/step
                                                                             - loss:    0.2304
- acc: 0.9189 - val_loss: 0.4140 - val_acc: 0.8728

Epoch 00012: val_acc did not improve from 0.93750Epoch 13/50
1824/1824 [==============================] - 20s 11ms/step
                                                                             - loss:    0.1787
- acc: 0.9430 - val_loss: 0.3403 - val_acc: 0.9107

Epoch 00013: val_acc did not improve from 0.93750Epoch 14/50
1824/1824 [==============================] - 19s 11ms/step
                                                                             - loss:    0.1734
- acc: 0.9419 - val_loss: 0.2575 - val_acc: 0.9286

Epoch 00014: val_acc did not improve from 0.93750Epoch 15/50
1824/1824 [==============================] - 19s 10ms/step
                                                                             - loss:    0.1523
- acc: 0.9501 - val_loss: 0.2354 - val_acc: 0.9330

Epoch 00015: val_acc did not improve from 0.93750Epoch 16/50
1824/1824 [==============================] - 19s 10ms/step
                                                                             - loss:    0.0997
- acc: 0.9649 - val_loss: 0.7065 - val_acc: 0.8616

Epoch 00016: val_acc did not improve from 0.93750Epoch 17/50
1824/1824 [==============================] - 18s 10ms/step
                                                                             - loss:    0.1160
- acc: 0.9644 - val_loss: 0.3953 - val_acc: 0.9263

Epoch 00017: val_acc did not improve from 0.93750Epoch 18/50
1824/1824 [==============================] - 18s 10ms/step
                                                                             - loss:    0.1124
- acc: 0.9660 - val_loss: 0.3622 - val_acc: 0.9286
```

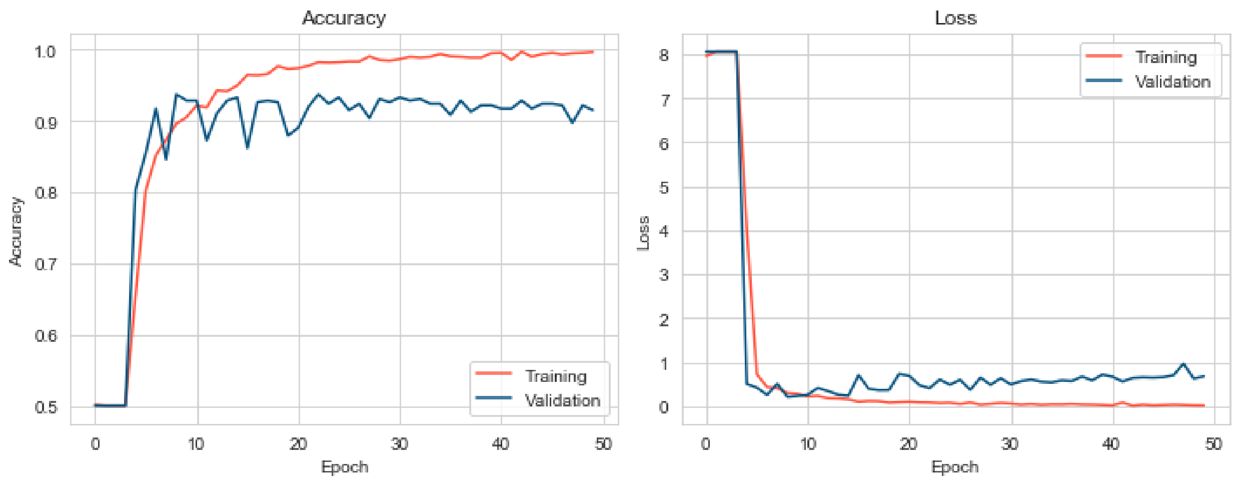
```
Epoch 00018: val_acc did not improve from 0.93750Epoch 19/50
1824/1824 [==============================] - 18s 10ms/step
                                                                         - loss:      0.0781
- acc: 0.9770 - val_loss: 0.3651 - val_acc: 0.9263

Epoch 00019: val_acc did not improve from 0.93750Epoch 20/50
1824/1824 [==============================] - 18s 10ms/step
                                                                         - loss:      0.0896
- acc: 0.9731 - val_loss: 0.7346 - val_acc: 0.8795

Epoch 00020: val_acc did not improve from 0.93750Epoch 21/50
1824/1824 [==============================] - 19s 10ms/step
                                                                         - loss:      0.0980
- acc: 0.9742 - val_loss: 0.6882 - val_acc: 0.8906

Epoch 00021: val_acc did not improve from 0.93750Epoch 22/50
1824/1824 [==============================] - 20s 11ms/step
                                                                         - loss:      0.0843
- acc: 0.9775 - val_loss: 0.4760 - val_acc: 0.9196

Epoch 00022: val_acc did not improve from 0.93750Epoch 23/50
1824/1824 [==============================] - 21s 12ms/step
                                                                         - loss:      0.0810
- acc: 0.9825 - val_loss: 0.4074 - val_acc: 0.9375

Epoch 00023: val_acc did not improve from 0.93750Epoch 24/50
1824/1824 [==============================] - 21s 12ms/step
                                                                         - loss:      0.0680
- acc: 0.9819 - val_loss: 0.6060 - val_acc: 0.9241

Epoch 00024: val_acc did not improve from 0.93750Epoch 25/50
1824/1824 [==============================] - 20s 11ms/step
                                                                         - loss:      0.0743
- acc: 0.9825 - val_loss: 0.4872 - val_acc: 0.9330

Epoch 00025: val_acc did not improve from 0.93750Epoch 26/50
1824/1824 [==============================] - 19s 10ms/step
                                                                         - loss:      0.0469
- acc: 0.9836 - val_loss: 0.6003 - val_acc: 0.9152

Epoch 00026: val_acc did not improve from 0.93750Epoch 27/50
1824/1824 [==============================] - 19s 10ms/step
                                                                         - loss:      0.0808
- acc: 0.9836 - val_loss: 0.3693 - val_acc: 0.9241

Epoch 00027: val_acc did not improve from 0.93750Epoch 28/50
1824/1824 [==============================] - 21s 11ms/step
                                                                        - loss:    0.0319

- acc: 0.9907 - val_loss: 0.6494 - val_acc: 0.9040

Epoch 00028: val_acc did not improve from 0.93750Epoch 29/50
1824/1824 [==============================] - 20s 11ms/step
                                                                        - loss:    0.0501

- acc: 0.9857 - val_loss: 0.4839 - val_acc: 0.9308

Epoch 00029: val_acc did not improve from 0.93750Epoch 30/50
1824/1824 [==============================] - 18s 10ms/step
                                                                        - loss:    0.0702

- acc: 0.9846 - val_loss: 0.6352 - val_acc: 0.9263

Epoch 00030: val_acc did not improve from 0.93750Epoch 31/50
1824/1824 [==============================] - 19s 10ms/step
                                                                        - loss:    0.0568

- acc: 0.9868 - val_loss: 0.4939 - val_acc: 0.9330

Epoch 00031: val_acc did not improve from 0.93750Epoch 32/50
1824/1824 [==============================] - 20s 11ms/step
                                                                        - loss:    0.0333

- acc: 0.9901 - val_loss: 0.5689 - val_acc: 0.9286

Epoch 00032: val_acc did not improve from 0.93750Epoch 33/50
1824/1824 [==============================] - 20s 11ms/step
                                                                        - loss:    0.0477

- acc: 0.9890 - val_loss: 0.6067 - val_acc: 0.9308

Epoch 00033: val_acc did not improve from 0.93750Epoch 34/50
1824/1824 [==============================] - 19s 10ms/step
                                                                        - loss:    0.0297

- acc: 0.9901 - val_loss: 0.5569 - val_acc: 0.9241

Epoch 00034: val_acc did not improve from 0.93750Epoch 35/50
1824/1824 [==============================] - 19s 10ms/step
                                                                        - loss:    0.0405

- acc: 0.9940 - val_loss: 0.5417 - val_acc: 0.9241

Epoch 00035: val_acc did not improve from 0.93750
Epoch 36/50
1824/1824 [==============================] - 19s 10ms/step                    - loss:    0.0387
- acc: 0.9907 - val_loss: 0.5860 - val_acc: 0.9085

Epoch 00036: val_acc did not improve from 0.93750Epoch 37/50
1824/1824 [==============================] - 19s 10ms/step
                                                                              - loss:    0.0485

- acc: 0.9901 - val_loss: 0.5715 - val_acc: 0.9286

Epoch 00037: val_acc did not improve from 0.93750Epoch 38/50
1824/1824 [==============================] - 18s 10ms/step
                                                                              - loss:    0.0366

- acc: 0.9890 - val_loss: 0.6733 - val_acc: 0.9129

Epoch 00038: val_acc did not improve from 0.93750Epoch 39/50
1824/1824 [==============================] - 18s 10ms/step
                                                                              - loss:    0.0337

- acc: 0.9890 - val_loss: 0.5863 - val_acc: 0.9219

Epoch 00039: val_acc did not improve from 0.93750Epoch 40/50
1824/1824 [==============================] - 19s 10ms/step
                                                                              - loss:    0.0256

- acc: 0.9951 - val_loss: 0.7194 - val_acc: 0.9219

Epoch 00040: val_acc did not improve from 0.93750Epoch 41/50
1824/1824 [==============================] - 19s 10ms/step
                                                                              - loss:    0.0151

- acc: 0.9956 - val_loss: 0.6697 - val_acc: 0.9174

Epoch 00041: val_acc did not improve from 0.93750Epoch 42/50
1824/1824 [==============================] - 18s 10ms/step
                                                                              - loss:    0.0786

- acc: 0.9857 - val_loss: 0.5607 - val_acc: 0.9174

Epoch 00042: val_acc did not improve from 0.93750Epoch 43/50
1824/1824 [==============================] - 18s 10ms/step
                                                                              - loss:    0.0083

- acc: 0.9973 - val_loss: 0.6423 - val_acc: 0.9286

Epoch 00043: val_acc did not improve from 0.93750Epoch 44/50
1824/1824 [==============================] - 21s 11ms/step
                                                                              - loss:    0.0325

- acc: 0.9901 - val_loss: 0.6561 - val_acc: 0.9174
```

```
Epoch 00044: val_acc did not improve from 0.93750
Epoch 45/50
1824/1824 [==============================] - 21s 11ms/step          - loss:    0.0154
- acc: 0.9940 - val_loss: 0.6484 - val_acc: 0.9241

Epoch 00045: val_acc did not improve from 0.93750
Epoch 46/50
1824/1824 [==============================] - 21s 11ms/step          - loss:    0.0235
- acc: 0.9956 - val_loss: 0.6600 - val_acc: 0.9241

Epoch 00046: val_acc did not improve from 0.93750
Epoch 47/50
1824/1824 [==============================] - 19s 11ms/step          - loss:    0.0320
- acc: 0.9934 - val_loss: 0.7059 - val_acc: 0.9219

Epoch 00047: val_acc did not improve from 0.93750
Epoch 48/50
1824/1824 [==============================] - 18s 10ms/step          - loss:    0.0259
- acc: 0.9951 - val_loss: 0.9661 - val_acc: 0.8973

Epoch 00048: val_acc did not improve from 0.93750
Epoch 49/50
1824/1824 [==============================] - 18s 10ms/step          - loss:    0.0169
- acc: 0.9956 - val_loss: 0.6273 - val_acc: 0.9219

Epoch 00049: val_acc did not improve from 0.93750
Epoch 50/50
1824/1824 [==============================] - 18s 10ms/step          - loss:    0.0154
- acc: 0.9967 - val_loss: 0.6788 - val_acc: 0.9152

Epoch 00050: val_acc did not improve from 0.93750
```

In [12]:

```
plot_metrics(d2_history1)
```

```
Best Model:
epoch: 9 , val_acc: 0.9375 , val_loss: 0.2073782096683447
```



Fine Tuning

In [ ]:

```
ft_model, ft_history = finetune_binary_model()Model
loaded.
Found 1824 images belonging to 2 classes.Found 448
images belonging to 2 classes. Epoch 1/50
114/114 [==============================] - 1571s 14s/step - loss: 0.3264 -
acc: 0.8799 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00001: val_acc improved from -inf to 0.93750, saving model todata2/ft_model.h5
Epoch 2/50
114/114 [==============================] - 1573s 14s/step - loss: 0.3228 -
acc: 0.8871 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00002: val_acc did not improve from 0.93750Epoch 3/50
114/114 [==============================] - 24835s 218s/step - loss: 0.3095
- acc: 0.8860 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00003: val_acc did not improve from 0.93750Epoch 4/50
114/114 [==============================] - 38145s 335s/step - loss: 0.3335
- acc: 0.8854 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00004: val_acc did not improve from 0.93750Epoch 5/50
114/114 [==============================] - 1602s 14s/step - loss: 0.3127 -
acc: 0.8942 - val_loss: 0.2074 - val_acc: 0.9375
```

```
Epoch 00005: val_acc did not improve from 0.93750Epoch 6/50
114/114 [==============================] - 1624s 14s/step - loss: 0.2912 -
acc: 0.8964 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00006: val_acc did not improve from 0.93750Epoch 7/50
114/114 [==============================] - 1649s 14s/step - loss: 0.3236 -
acc: 0.8843 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00007: val_acc did not improve from 0.93750Epoch 8/50
114/114 [==============================] - 1711s 15s/step - loss: 0.3301 -
acc: 0.8887 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00008: val_acc did not improve from 0.93750Epoch 9/50
110/114 [===========================>..] - ETA: 1:06 - loss: 0.3238 - acc:
0.8841
```

In [ ]:

```
plot_metrics(ft_history)
```

Load Model

In [17]:

```
ft_model = load_model(location+'/ft_model.h5') WARNING:tensorflow:From
C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in afuture version.
Instructions for updating:Use tf.cast
instead.
C:\Anaconda3\envs\envdlcv\lib\site-packages\keras\engine\saving.py:327: UserWarning: Error in
loading the saved optimizer state. As a result, yourmodel is starting with a freshly initialized
optimizer.
   warnings.warn('Error in loading the saved optimizer '
```

In [ ]:

```
with open('data1a/top_history.txt') as f:top_history =
    json.load(f)
```

In [ ]:

```
with open('data1a/ft_history.txt') as f:ft_history =
    json.load(f)
```

In [ ]:

```
plot_acc_metrics(top_history, ft_history)
```

In [22]:

```
validation_labels = np.array([0] * (nb_validation_samples // 2) + [1] *
(nb_validation_samples // 2))
```

In [51]:

```
cm = evaluate_binary_model(ft_model, validation_data_dir,
validation_labels)
```

```
Found 448 images belonging to 2 classes.
  ---------------------------------------------------------------------------------------------------------
  -



  KeyboardInterrupt                                Traceback (most recent calllast)
  <ipython-input-51-bf52512d511d> in <module>
----> 1 cm = evaluate_binary_model(ft_model, validation_data_dir,validation_labels)

  <ipython-input-27-304db6f68ef2> in evaluate_binary_model(model, directory,labels)
            4          generator = datagen.flow_from_directory(directory,
  target_size=(img_height,img_width), batch_size=batch_size, class_mode='binary',
  shuffle=False)
            5
  ----> 6    predictions = model.predict_generator(generator, len(labels))7
            8          pred_labels = [0 if i<0.5 else 1 for i in predictions]

 C:\Anaconda3\envs\envdlcv\lib\site-
 packages\keras\legacy\interfaces.py inwrapper(*args, **kwargs)
          89                              warnings.warn('Update your `' + object_name + '`
call to the ' +
          90                                                  'Keras 2 API: ' + signature,
stacklevel=2)
  ---> 91                          return func(*args, **kwargs)
          92                 wrapper._original_function = func
          93                 return wrapper

 C:\Anaconda3\envs\envdlcv\lib\site-
 packages\keras\engine\training.py inpredict_generator(self,
 generator, steps, max_queue_size, workers, use_multiprocessing,
 verbose)
       1520                         workers=workers,
       1521                         use_multiprocessing=use_multiprocessing,
-> 1522                         verbose=verbose)
```

```
C:\Anaconda3\envs\envdlcv\lib\site- packages\keras\engine\training_generator.py in
predict_generator(model, generator, steps, max_queue_size, workers, use_multiprocessing, verbose)
      451                         x = generator_output452
--> 453                     outs = model.predict_on_batch(x)
      454                     outs = to_list(outs)455

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\engine\training.py inpredict_on_batch(self, x)
     1272                   ins = x
     1273           self._make_predict_function()
-> 1274             outputs = self.predict_function(ins)1275    return
     unpack_singleton(outputs)
     1276

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py in__call__(self,
inputs)
     2713                       return self._legacy_call(inputs)
     2714

-> 2715                   return self._call(inputs)2716
            else:
     2717                 if py_any(is_tensor(x) for x in inputs):

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py in _call(self,
inputs)
    2673                  fetched = self._callable_fn(*array_vals,
run_metadata=self.run_metadata)
     2674           else:
-> 2675                  fetched = self._callable_fn(*array_vals)2676 return
     fetched[:len(self.outputs)]
     2677

C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\client\session.py in__call__(self,
*args,
**kwargs)
     1437              ret = tf_session.TF_SessionRunCallable(
     1438                  self._session._session, self._handle, args, status,
-> 1439                         run_metadata_ptr)
   1440                  if run_metadata:
   1441                    proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)
```

```
KeyboardInterrupt:
```

In [ ]:

```
heatmap_laebls = ['Damaged', 'Whole']
```

In [ ]:

```
sns.heatmap(cm, annot=True, annot_kws={"size":16}, fmt='g', cmap='OrRd',
xticklabels=heatmap_labels, yticklabels=heatmap_labels)
```

In [ ]:

```
sns.heatmap(cm, annot=Ture, annot_kws={"size":16}, fmt='g', cmap='Blues',
xticklabels=heatmap_labels, yticklabels=heatmap_labels)
```

Pipe2

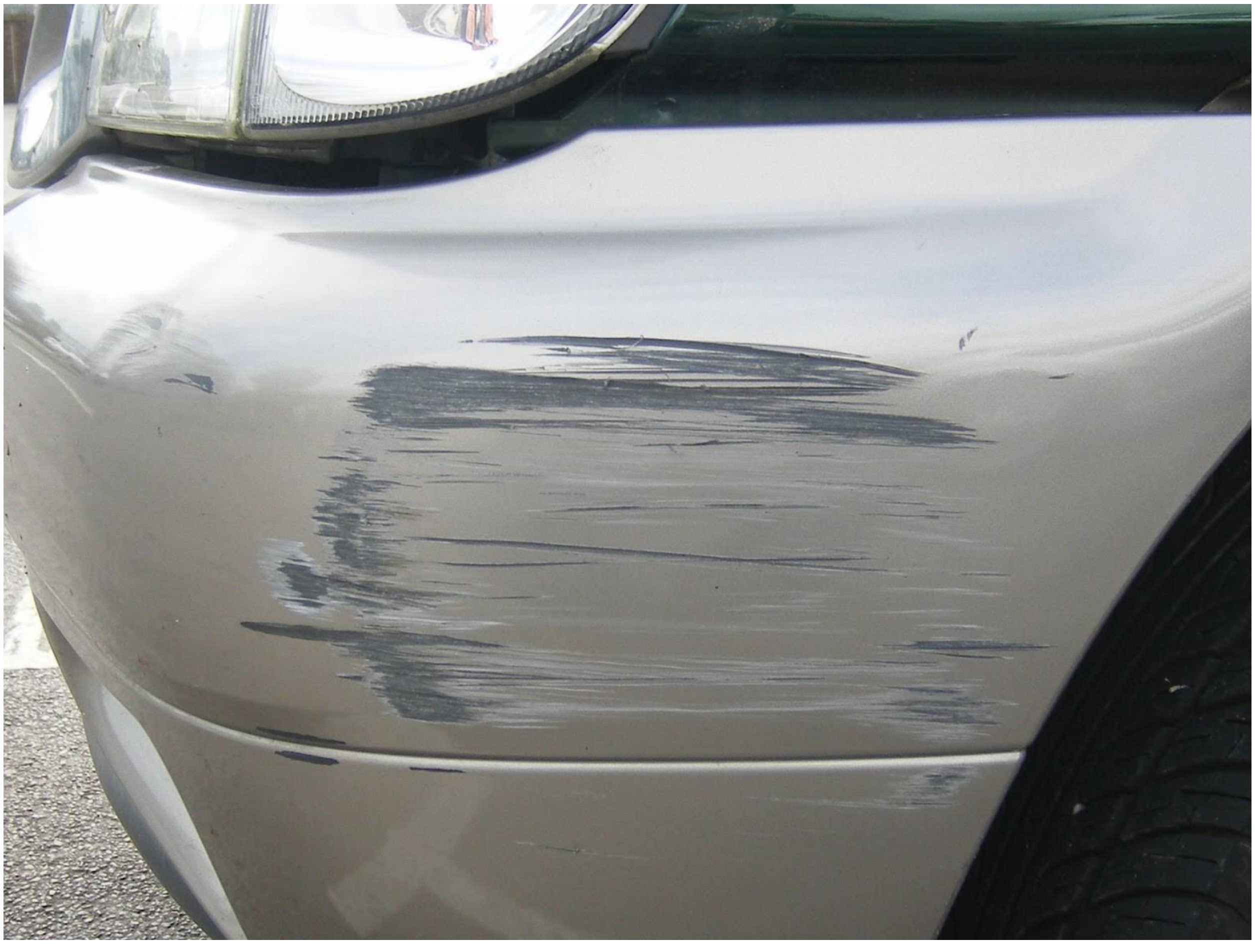
In [11]:

```
  def pipe2(image_path, model): urllib.request.urlretrieve(image_path,
        'save.jpg')img = load_img('save.jpg', target_size=(256,256)) x =
        img_to_array(img)
        x = x.reshape((1,) + x.shape)/255pred =
        model.predict(x)
        print("Validating that damage exists........................ ")
        print(pred)
        if(pred[0][0]<=0.5):
            print("Validation complete - proceed to location and severitydetermination")
        else:
            print ("Are you sure that your car is damaged? Please submitanother picture of
the damage.")
            print ("Hint: Try zooming in/out, using a different angle ordifferent lighting")
```

In [12]:

```
Image('http://3.bp.blogspot.com/-
PrRY9XxCqYQ/UDNutnMI7LI/AAAAAAAABdw/UGygghh-hRA/s1600/Bumper+scuff.JPG')
```

Out[12]:



In [18]:

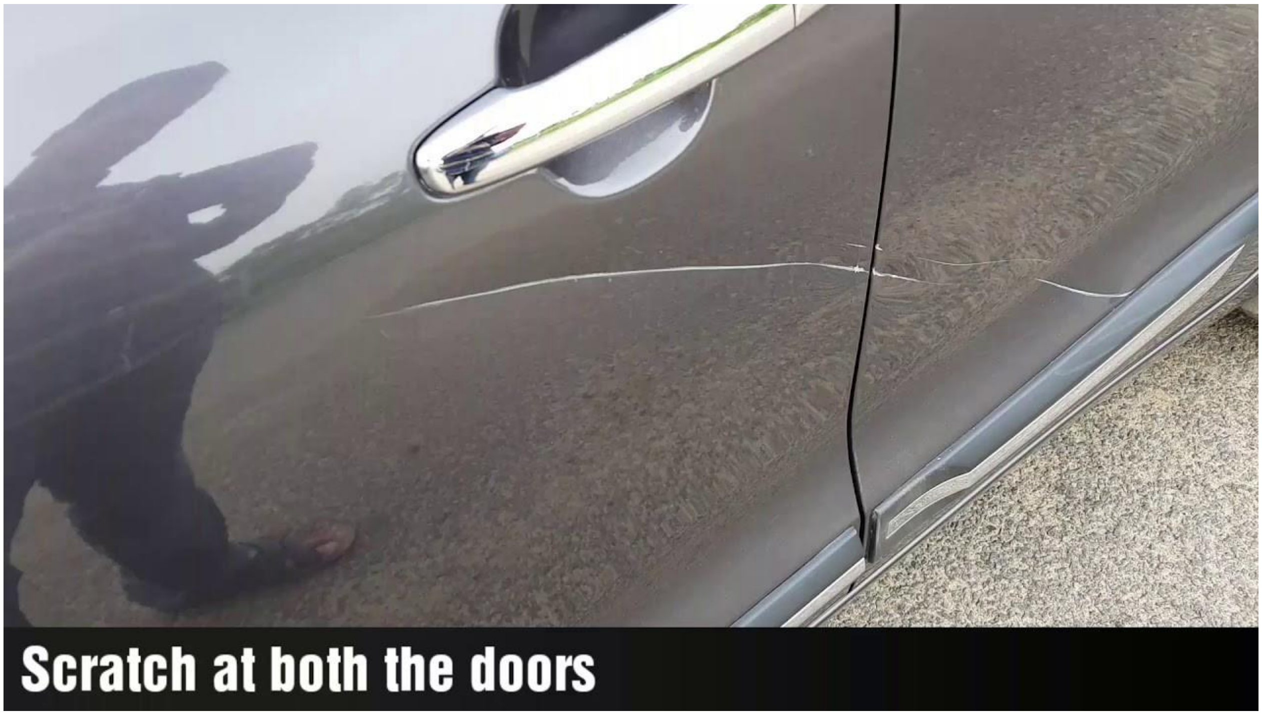
```
pipe2('http://3.bp.blogspot.com/-PrRY9XxCqYQ/UDNutnMI7LI/AAAAAAAABdw/UGygghh-hRA/s1600/Bumper+scuff.JPG', ft_model)
```

```
Validating that damage exists....
[[0.0002488]]
Validation complete - proceed to location and severity determination
```

In [40]:

```
Image('https://i.ytimg.com/vi/4oV1klVPogY/maxresdefault.jpg')
```

Out[40]:



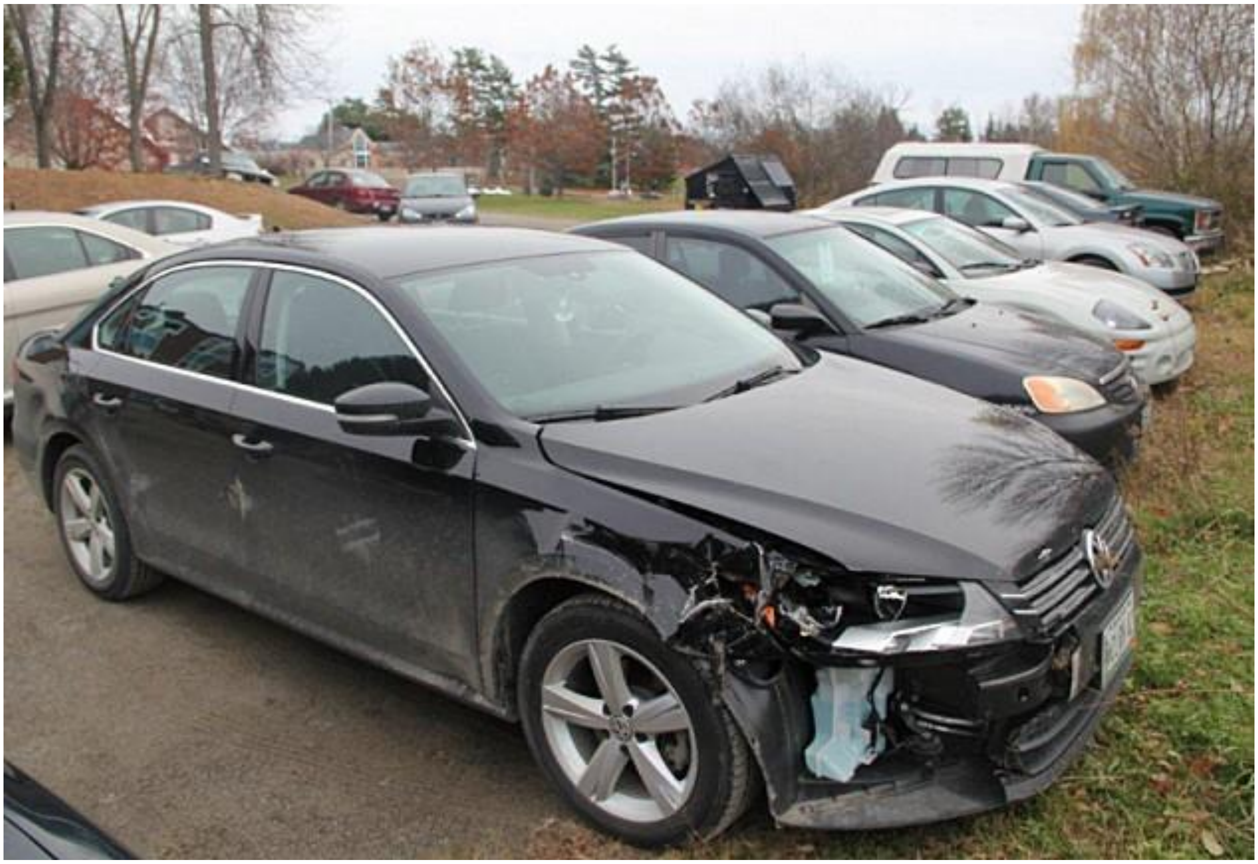
In [41]:

```
pipe2('https://i.ytimg.com/vi/4oV1klVPogY/maxresdefault.jpg', ft_model)Validating that damage exists....
[[0.01300194]]
Validation complete - proceed to location and severity determination
```

In [47]:

```
Image('http://blog.automart.co.za/wp-content/uploads/2014/09/Accident_Damaged_Car.png')
```

Out[47]:



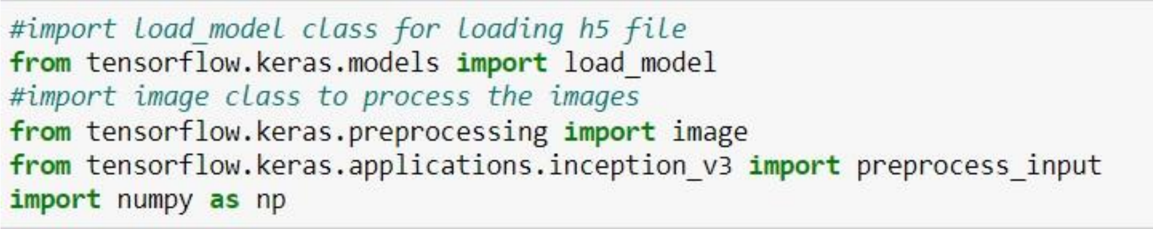
In [46]:

```
pipe2('http://blog.automart.co.za/wp-
content/uploads/2014/09/Accident_Damaged_Car.png', ft_model)Validating
that damage exists....
[[0.11757535]]
Validation complete - proceed to location and severity determination
```

8.TESTING

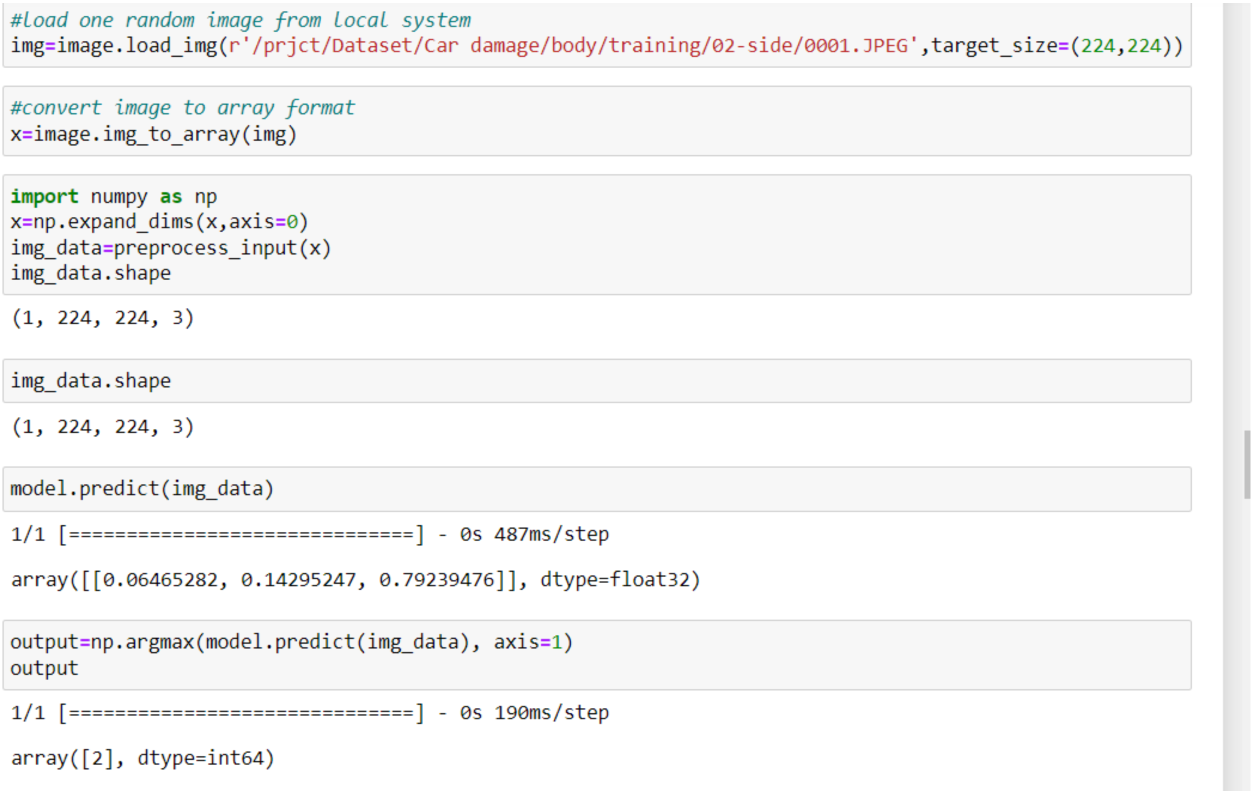
8.1 Test Cases:

Evaluation is a process during the development of the model to check whether the model is the bestfit for the given problem and corresponding data.
Load the saved model using load_model



```
#import load_model class for loading h5 file
from tensorflow.keras.models import load_model
#import image class to process the images
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
```

Taking an image as input and checking the results



```
#load one random image from local system
img=image.load_img(r'/prjct/Dataset/Car damage/body/training/02-side/0001.JPEG',target_size=(224,224))
```

```
#convert image to array format
x=image.img_to_array(img)
```

```
import numpy as np
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
img_data.shape
```

```
(1, 224, 224, 3)
```

```
img_data.shape
```

```
(1, 224, 224, 3)
```

```
model.predict(img_data)
```

```
1/1 [==============================] - 0s 487ms/step

array([[0.06465282, 0.14295247, 0.79239476]], dtype=float32)
```

```
output=np.argmax(model.predict(img_data), axis=1)
output
```

```
1/1 [==============================] - 0s 190ms/step

array([2], dtype=int64)
```

8.2 User Acceptance Testing:

8.2.1 Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies project at the time of the release to User Acceptance Testing (UAT).

8.2.2 Defect Analysis:

This report shows the number of resolved or closedbugs at eachseverity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	24	14	13	26	77

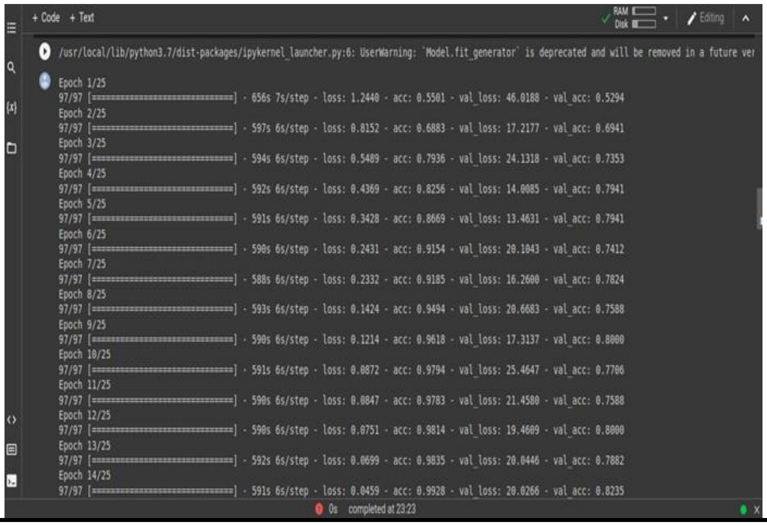
8.2.3 Test Case Analysis:

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	50	0	0	50
Security	4	0	0	4
Outsource Shipping	3	0	0	3

Exception Reporting	8	0	0	8
Final Report Output	5	0	0	5
Version Control	2	0	0	2

9.Results:

S.No.	Parameter	Values
1.	Model Summary	 /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future ver Epoch 1/25 97/97 [==============================] - 656s 7s/step - loss: 1.2440 - acc: 0.5501 - val_loss: 46.0188 - val_acc: 0.5294 Epoch 2/25 97/97 [==============================] - 597s 6s/step - loss: 0.8152 - acc: 0.6883 - val_loss: 17.2177 - val_acc: 0.6941 Epoch 3/25 97/97 [==============================] - 594s 6s/step - loss: 0.5489 - acc: 0.7936 - val_loss: 24.1318 - val_acc: 0.7353 Epoch 4/25 97/97 [==============================] - 592s 6s/step - loss: 0.4369 - acc: 0.8256 - val_loss: 14.0085 - val_acc: 0.7941 Epoch 5/25 97/97 [==============================] - 591s 6s/step - loss: 0.3428 - acc: 0.8669 - val_loss: 13.4631 - val_acc: 0.7941 Epoch 6/25 97/97 [==============================] - 590s 6s/step - loss: 0.2431 - acc: 0.9154 - val_loss: 20.1043 - val_acc: 0.7412 Epoch 7/25 97/97 [==============================] - 588s 6s/step - loss: 0.2332 - acc: 0.9185 - val_loss: 16.2600 - val_acc: 0.7824 Epoch 8/25 97/97 [==============================] - 593s 6s/step - loss: 0.1424 - acc: 0.9494 - val_loss: 20.6683 - val_acc: 0.7588 Epoch 9/25 97/97 [==============================] - 590s 6s/step - loss: 0.1214 - acc: 0.9618 - val_loss: 17.3137 - val_acc: 0.8000 Epoch 10/25 97/97 [==============================] - 591s 6s/step - loss: 0.0872 - acc: 0.9794 - val_loss: 25.4647 - val_acc: 0.7706 Epoch 11/25 97/97 [==============================] - 590s 6s/step - loss: 0.0847 - acc: 0.9783 - val_loss: 21.4580 - val_acc: 0.7588 Epoch 12/25 97/97 [==============================] - 590s 6s/step - loss: 0.0751 - acc: 0.9814 - val_loss: 19.4609 - val_acc: 0.8000 Epoch 13/25 97/97 [==============================] - 592s 6s/step - loss: 0.0699 - acc: 0.9835 - val_loss: 20.0446 - val_acc: 0.7882 Epoch 14/25 97/97 [==============================] - 591s 6s/step - loss: 0.0459 - acc: 0.9928 - val_loss: 20.0266 - val_acc: 0.8235 0s completed at 23:23
2.	Accuracy	Training Accuracy for body model- 98.6% Validation Accuracy for body model- 67% Training Accuracy for level model - 99.79% Validation Accuracy for level model - 62% -

10.ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

Thanks to digitisation, the claim process is simple to use.

Conduct a comprehensive analysis of the damaged vehicle.

Helps in the analysis of the damaged car and the payment process by theinsurance company

.

DISADVANTAGES:

The manual method for submitting an insurance claim will take longer.

The corporation acts improperly and currently doesn't make payments as aresult of false accusations.

Poor customer service.

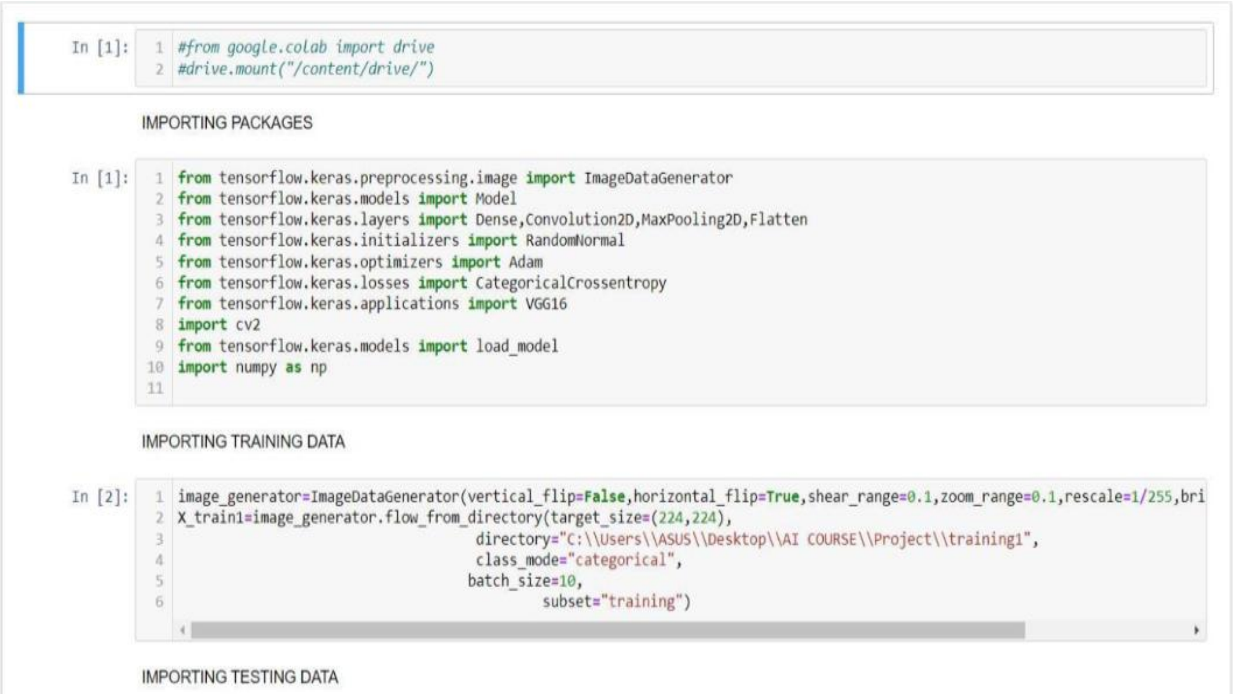
11.CONCLUSION:

In this research proposal, an automotive detection approach based on neural networks will be used to address the issues of car damage analysis and position and severity prediction. This project completes several tasks at once. Undoubtedly, the method will assist the insurance firms in conducting far more thorough and systematic examinations of the vehicle damage. The technology can evaluate a snapshot of the car to determine whether damage is present, where it is located, and how severe it is.

12.FUTURE SCOPE:

In our upcoming work, we'll need to employ numerous regularisation methods and a sizable dataset. We can more accurately and reliably estimate the cost of a broken automotive component if we have higher quality datasets that include the characteristics of a car (make, model, and year of manufacture), location information, the type of damaged part, and repair cost. This study prepares the pathfor future photo recognition initiatives with a focus on the auto insurance industry. The study was able to validate the existence of damage, its location, and its degree with accuracy by eliminating human bias. They can be further enhanced by incorporating the on-the-fly data augmentation approach.

13.APPENDIX:



In [1]:

```
#from google.colab import drive
#drive.mount("/content/drive/")
```

IMPORTING PACKAGES

In [1]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
from tensorflow.keras.initializers import RandomNormal
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.applications import VGG16
import cv2
from tensorflow.keras.models import load_model
import numpy as np

```

IMPORTING TRAINING DATA

In [2]:

```
image_generator=ImageDataGenerator(vertical_flip=False,horizontal_flip=True,shear_range=0.1,zoom_range=0.1,rescale=1/255,bri
X_train1=image_generator.flow_from_directory(target_size=(224,224),
                                     directory="C:\\Users\\ASUS\\Desktop\\AI COURSE\\Project\\training1",
                                     class_mode="categorical",
                                    batch_size=10,
                                             subset="training")
```

IMPORTING TESTING DATA

GitHub Link: **IBM-Project-12410-1659450816**

Project Demo Link:

https://drive.google.com/file/d/1xMZNdUUtdlO7aAYNSIdy-qVW6wXMqFr3/view?usp=share_link