A PROJECT REPORT

on

"CAR DAMAGE IDENTIFICATION MODEL"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

CHIRAYATA CHATERJI	1705779
PRIYOTOSH SAHA	1705869
RISHIT DUBEY	1705768
SHUBHAM BANERJEE	1705825
SWAYAMDIPTA SAHA	1705837

UNDER THE GUIDANCE OF
PROF. BISWAJIT SAHOO

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2020

A PROJECT REPORT

on

"CAR DAMAGE IDENTIFICATION MODEL"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN

COMPUTER SC. AND ENGINEERING

BY

CHIRAYATA CHATTERJI	1705779
PRIYOTOSH SAHA	1705869
RISHIT DUBEY	1705768
SHUBHAM BANERJEE	1705825
SWAYAMDIPTA SAHA	1705837

UNDER THE GUIDANCE OF

PROF. BISWAJIT SAHOO

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024

May 2020

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"CAR DAMAGE IDENTIFICATION MODEL"

submitted by

CHIRAYATA CHATTERJI	1705779
PRIYOTOSH SAHA	1705869
RISHIT DUBEY	1705768
SHUBHAM BANERJEE	1705825
SWAYAMDIPTA SAHA	1705837

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. Biswajit Sahoo)
Project Guide

Acknowledgement

We are profoundly grateful to Prof. BISWAJIT SAHOO for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. The work is a team effort minus which the completion of this project was not possible.

CHIRAYATA CHATTERJI
PRIYOTOSH SAHA
RISHIT DUBEY
SHUBHAM BANERJEE
SWAYAMDIPTA SAHA

ABSTRACT

Recent advances in deep learning and computation infrastructure(cloud,GPUs etc.) have made computer vision applications leap forward: from unlocking office access door with our face to self-driving cars.Our project does something like that.In this project convolution neural networks are used to recognize whether a car on a given image is damaged or not. Using transfer learning to take the advantage of available models that are trained on a more general object recognition task,very satisfactory performances have been achieved,which indicate the great opportunities of this approach.

In the end,a promising attempt is made in classifying car damages into a few different classes is presented. Along the way, the main focus has been on the influence of certain hyper-parameters and on seeking theoretically founded ways to adapt them, all with the objective of progressing to satisfactory results as fast as possible. This problem-solving approach open doors for future collaborations on image recognition projects in general and for the car insurance field in particular.

Keywords: Tranfer Learning, Convolution Neural Network,Deep learning,,Mask-RCNN,Hyperparameter-Tuning

Contents

1 Introduction 1

1.1 Brief
1.2 Current System
1.2.1 Data Flow Diagram
1.3 Objective

2 Literature Survey 3

2.1 Related Works

3 Software Requirements Specification 4

3.1 Initial Specifications
3.1.1 Functional Specifications
3.1.2 Non-Functional Specifications
3.2 Software/Packages Specifications
3.2.1 Functional Specifications

4 Requirement Analysis 6

4.1 Hardware Development
4.2 Software Requirements
4.3 Cost/Benefit Analysis
4.3.1 Cost Analysis
4.3.2 Effort Analysis

5 System Design 8

5.1 Architectural View Decomposition Procedural Design
5.2 Use-Case Diagram
5.3 Activity Diagram

6 Definitions and Methodology 10
6.1 Convolutional Neural Networks
6.2 Mask R-CNN

6.3 Feature Extraction
6.3.1 Convolution Layer
6.3.2 Non-Linearity
6.3.3 Pooling Layer

6.4 Classification Part

6.5 Transfer Learning

6.5.1 Re-purposing a Pre-Trained Model

6.6 VGG16

7 Project Planning 15

7.1 Data Collection
7.2 Data Annotation
7.3 Environment Set-Up
7.4 Network Training
7.5 Model Validation
7.6 Model Prediction

8 Implementation 17

9 Screen Shots of Project 26
9.1 Homepage
9.2 Service Page
9.3 Results Page
9.4 Homepage

10 Conclusion and Future Scope 29
10.1 Conclusion
10.2 Future Scope

11 References 30
12 Individual Reports 31

Chapter 1

INTRODUCTION

1.1 BRIEF

Not a many years ago image classification task, such as handwritten digit recognition(the great MNIST dataset) or basic object (cat/dog) identification was considered as a great success in the computer vision domain. However Convolution neural networks (CNN), the driver behind computer vision applications, are fast evolving with advanced and innovative architectures to solve almost any problem under the sun related to the visual system.

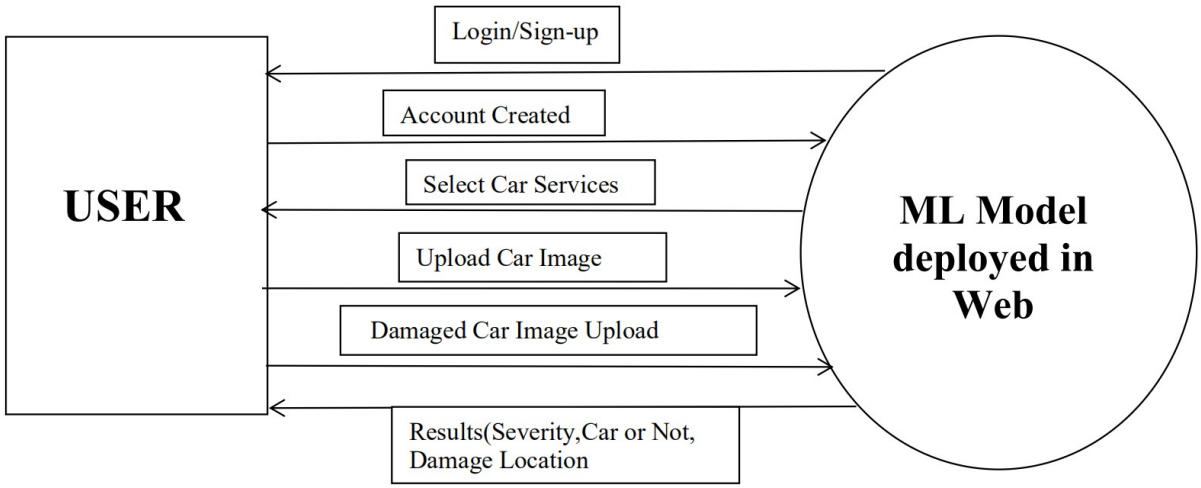
Automated detection of car exterior damages and subsequent quantification(damage severity) of those would help used car dealers(Marketplace) to price cars accurately and fast by eliminating the manual process of damage assessment. The concept is equally beneficial for property and casualty(P&C) insurers, in terms of faster claim settlement and hence greater customer satisfaction. In this article, We will step by step to describe the concept of car scratch(most frequent exterior damage) detection using CNN transfer learning leveraging Tensorflow back-end.

1.2 CURRENT SYSTEM

The applications of computer vision continue to amaze us. From detecting objects in a video, to counting the number of people in a crowd, there is no challenge that computer vision seemingly cannot overcome. One of the more intriguing applications of computer vision is identifying pixels in a scene and using them for diverse and remarkably useful purposes. We will be taking up one such application in this article, and trying to understand how it works using Python!.A customer or an Account holder has to frequently visit the Bank to claim insurance for car.The brochures and the verbal communication are the major way of getting information about the Co-Operative.Different software is used for different purpose. There is no one program that can handle the multitasking operations. Besides customer stand up in the line and wait for their

turn. Large numbers of staff are appointed to handle the customer. The basic concept of current system can be portrait with the help of following Data Flow Diagram.

1.2.1 DATA FLOW DIAGRAM



1.3 OBJECTIVE

The aim of this project is to build a custom Mask R-CNN model that can detect the area of damage on a car (see the image example below). 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 they can assess damage from them. This model can also be used by lenders if they are underwriting a car loan especially for a used car.

Chapter 2

LITERATURE SURVEY

2.1 RELATED WORKS

Deep learning has shown promising results in machine learning applications. In particular, CNNs perform well for computer vision tasks such as visual object recognition and detection. Application of CNNs to structural damage assessment has been studied in.

The authors propose a deep learning based method for Structural Health Monitoring (SHM) to characterize the damage in the form of cracks on a composite material. Unsupervised representation is employed and results have been shown on a wide range of loading conditions with limited number of labeled training image data. Most of the supervised methods need large amounts of labeled data and compute resources. Unsupervised pre-training techniques such as Auto-encoders have been proven to improve the generalization performance of the classifier in case of small number of labeled samples. For images, Convolutional Auto Encoders (CAE) have shown promising results.

A very well known technique which has worked effectively in case of small labeled data is transfer learning. A network which is trained on a source task is used as a feature extractor for target task. There are many CNN models trained on Imagenet which are available publicly such as VGG-16, VGG-19,Alexnet, Inception,Cars,Resnet.Transferable feature representation learned by CNN minimizes the effect of over-fitting in case of a small labeled set.Traditional machine learning techniques have also been experimented for automated damage assessment.

Jaywardena et al proposed a method for vehicle scratch damage detection by registering 3D CAD model of undamaged vehicle(ground truth) on the image of the damaged vehicle. There has been attempts to analyze damage in geographical regions using satellite images.To best of our knowledge,deep learning based techniques have not been employed for automated car damage classification, especially for the fine-granular damage classification.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 INITIAL SPECIFICATIONS

3.1.1 FUNCTIONAL SPECIFICATIONS

- The system should provide the interface for general users customers or members.
- The system should be able to carry out the simple image prediction.
- A user login system should be included.
- Customer should be able to print out the final results shown whenever they need.
- The system should be such that car's image can be regularly updated every time.
- The system should provide for the secure, reliable and efficient functioning.
- The system must be able to run with the internet efficiently.

3.1.2 NON-FUNCTIONAL SPECIFICATIONS

- User friendly interface.
- The system must be robust and secure enough from the unauthorized access and loss of data.
- A common platform for all the data processing.
- The system should meet the standard of the organization.

3.2 SOFTWARE/PACKAGES SPECIFICATIONS

3.1.2 FUNCTIONAL SPECIFICATIONS

This project uses the following tools/languages used for its implementation:

- **Python:** The programming language used to implement this project.
- **Jupyter Notebook:** Used to provide an interactive environment for python, for the implementation of this project.

- **NumPy:** NumPy is a library for the Python programming language, adding support for large, muti-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Keras:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.
- **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
- **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **Scikit-Learn:** It features various regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- **Anaconda** : Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.
- **Django:** Django is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern.
- **Visual Studio:** Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
- **HTML:** Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.
- **CSS:** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- **Javascript:** JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
- **Dialogflow**: **Dialogflow** (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations.

Chapter 4

REQUIREMENT ANALYSIS

4.1 HARDWARE REQUIREMENTS

FOR DEVELOPMENT:

Minimum hardware requirement:

- Pentium III with 1.66 GHz processor
- 512 Mb of RAM,
- 14 inch color monitor.

Tools requirement:

Web Browser, Web Server (XAMPP recommended),Anaconda,Jupyter Notebook.

Programming /Scripting Languages to be used:

Python, HTML, CSS, JAVASCRIPT.

FOR OPERATION OF SYSTEM:

Minimum Hardware Requirement:

- Pentium III with 1.66 GHz processor or higher,
- 512 Mb of RAM or higher,
- 14 inch color monitor.

4.2 SOFTWARE REQUIREMENTS

- O/S: Microsoft Windows 98, NT ME, 2000, XP (service packs 1, 2, or, 3),VISTA or Windows 7 also compatible with Linux and MAC OS X...
- Web Server: XAMPP(APACHE included)
- Browser: Minimum Microsoft Internet explorer 7 or higher, Opera 6.x, Firefox or Netscape 6.x, with JavaScript 1.3,Chrome
- Server Requirement: Web Server
  - Database Server

4.3 COST/ BENEFIT ANALYSIS

4.3.1 Cost Analysis

Initial cost =

- Software – Licensed Software Cost

4.3.2 Effort Analysis

Person Hours: 5 persons *135(9 hours * 5 days * 3 weeks)=675 hours

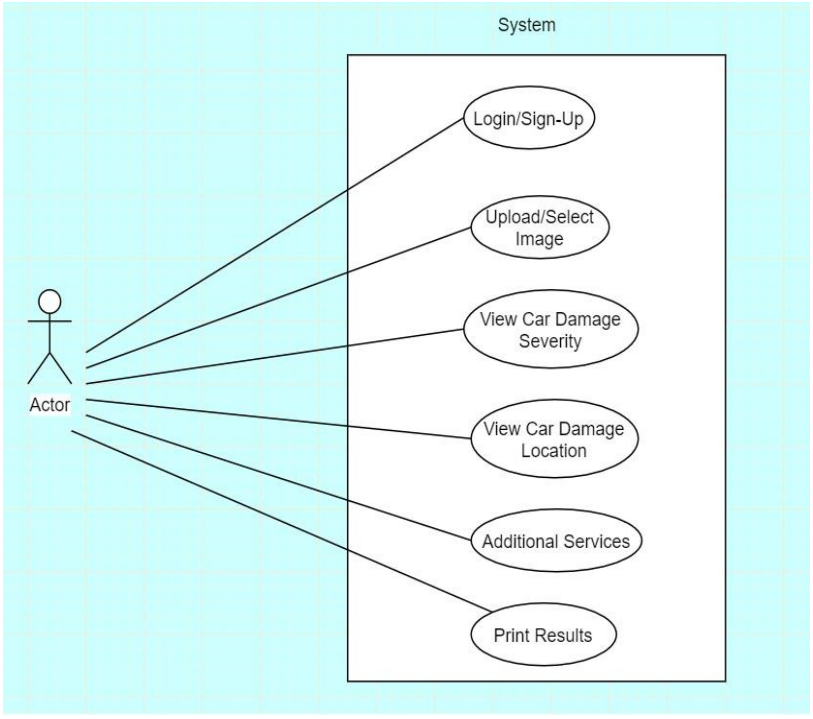
Chapter 5

SYSTEM DESIGN

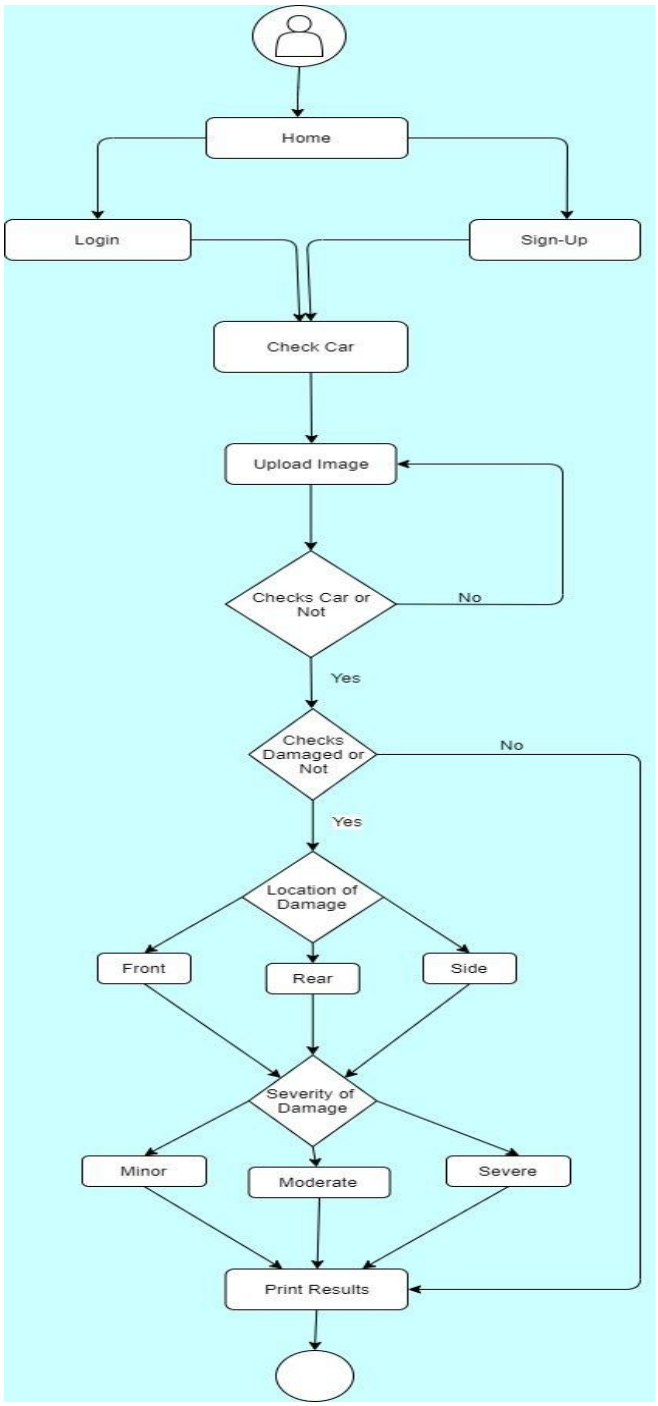
5.1 Architectural View Decomposition Procedural Design

The procedural design transforms the structural design of the system into the procedural description. There are several ways to represents the procedural details. Some programmers may use English like phrases, some may use graphical notations or some others may use the tabular design notation. In our case we use Graphical Design Notation i.e. Program flows to transform the structural modules into procedural description.

5.2 Use-Case Diagram



5.3 **Activity Diagram**



Chapter 6

DEFINITIONS AND METHODOLOGY

6.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning algorithm that can recognize and classify features in images for computer vision. It is a multi-layer neural network designed to analyze visual inputs and perform tasks such as image classification, segmentation and object detection etc. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field.

Historical Background

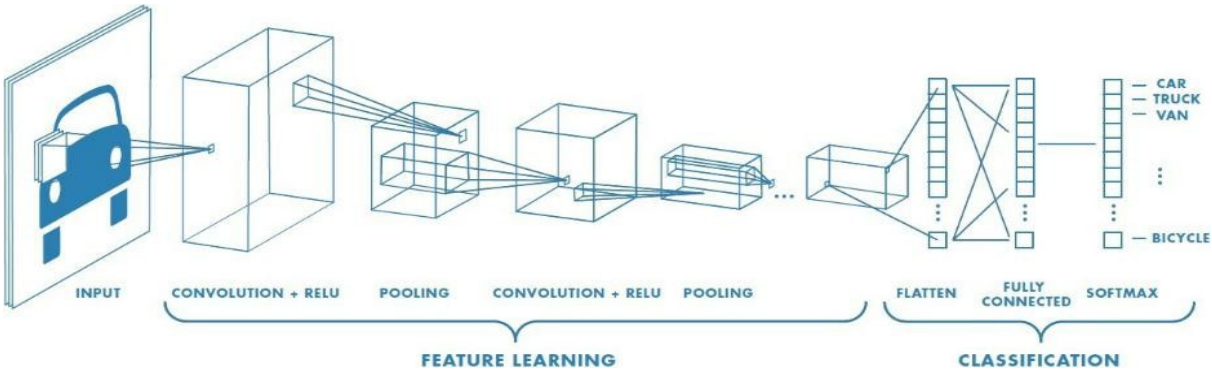
CNNs was popularized mostly thanks to the efforts of Yann LeCun, now the Director of AI Research at Facebook. In the early 1990s, LeCun worked at Bell Labs, one of the most prestigious research labs in the world at the time, and built a check-recognition system to read handwritten digits. There's a very interesting video from 1993 where LeCun showed how the system works right here. This system was actually an entire process for doing end-to-end image recognition. The resulting paper, that he co-authored with Leon Bottou, Patrick Haffner, and Yoshua Bengio in 1998, introduces convolutional nets as well as the full end-to-end system they built.

Key Features:-

- It needed to be able to detect the objects in the image and place them into the appropriate category.
- It also needed to be robust against differences in pose, scale, illumination, confirmation and clutter.
- Images are usually in grayscale or in color(RGB) format .The computers perceive the images as a matrix of pixel values having varying intensity values 0 to 255.
- A grayscale image, can be represented by a single matrix having pixel intensities from 0 to 255.
- A Color image, can be represented by three matrix having pixel intensities from 0 to 255,each matrix representing color intensities in RGB format.10

CNNs Components:-

- The Hidden layers/Feature extraction part: In this part, the network will perform a series of convolutions and pooling operations during which the features are detected.
- The Classification part : Here, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

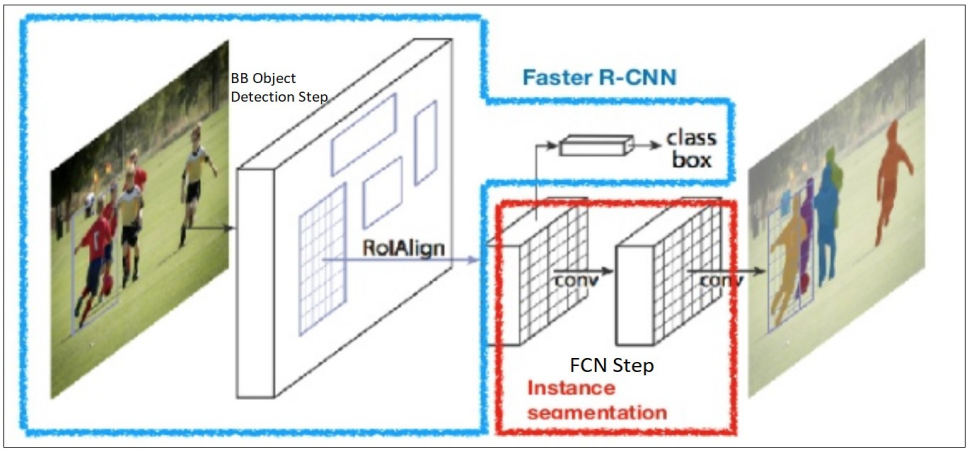


6.2 **Mask R-CNN**

Mask R-CNN is an instance segmentation model that allows identifying pixel-wise delineation for object class of our interest. So Mask R-CNN has two broad tasks-

1)BB based Object detection(also called localization task) and

2) Semantic segmentation, which allows segmenting individual objects at pixel within a scene,irrespective of the shapes.Put together these two tasks Mask R-CNN does get Instance Segmentation for a given image.



6.3 **Feature Extraction**

6.3.1 Convolution Layer

Convolution is a function that can detect edges,orientations, and small patterns in an image.

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

Here we extract features from the input image:

- We preserve the spatial relationship between pixels by learning image features using small squares of input data. These squares of input data are also called *filters or kernels*.
- The matrix formed by sliding the filter over the image and computing the dot product is called a *Feature Map*. The more filters we have, the more image features get extracted and the better our network becomes at recognizing patterns in unseen images.
- The size of our feature map is controlled by *depth* (the number of filters used), *stride* (the number of pixels slid over the input matrix), and *zero -padding* (padding the input matrix with 0s around the border).

Strides : Stride is the number of pixels shifts over the input matrix.
Padding: Sometimes filter does not fit perfectly fit the input image. We have two options:

- Pad the picture with zeros (zero-padding) so that it fits
- Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

6.3.1 Non-Linearity

For any kind of neural network to be powerful, it needs to contain non-linearity. LeNet uses *sigmoid non-linearity*, which takes a real-valued number and squashes it into a range between 0 and 1. In particular, large negative numbers become 0 and large positive numbers become 1. However, the sigmoid non-linearity has a couple of major drawbacks: (i) sigmoids saturate and kill gradients, (ii) sigmoids have slow convergence, and (iii) sigmoid outputs are not zero-centered.

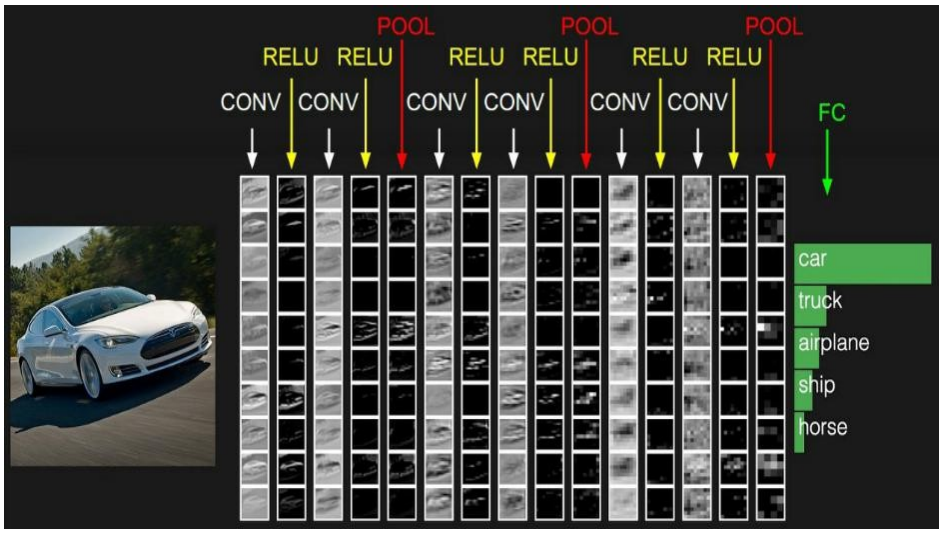
6.3.1 Pooling Layer

After this, we perform a *pooling* operation to reduce the dimensionality of each feature map. This enables us to reduce the number of parameters and computations in the network, therefore controlling overfitting.

- There are mainly three types of pooling:- Max Pooling,Mean Pooling, Sum Pooling

6.4 **Classification Part**

After the convolution and pooling layers, we add a couple of fully-connected layers to wrap up the CNN architecture. The output from the convolution and pooling layers represent high-level features of the input image. The FC layers use these features for classifying the input image into various classes based on the training dataset. Apart from classification, adding FC layers also helps to learn non-linear combinations of these features.



After applying these we will be able to detect whether the image is a car or not.

6.5 **Transfer Learning**

Transfer learning is a popular method in computer vision because it allows us to build accurate models in a time saving way. With transfer learning, instead of starting the learning process from scratch, you start from patterns that have been learned when solving a different problem.
In computer vision, transfer learning is usually expressed through the use of **pre-trained models**. From a practical perspective, the entire transfer learning process can be summarized as follows:

1. Select a pre-trained model.
2. Classify your problem according to the Size-Similarity Matrix.
3. Fine-tune your model.

6.5 **Re-purposing a Pre-trained Model**

We start by removing the original classifier, then we added a new classifier that fits your purposes, and finally we have to fine-tune your model according to one of three strategies:

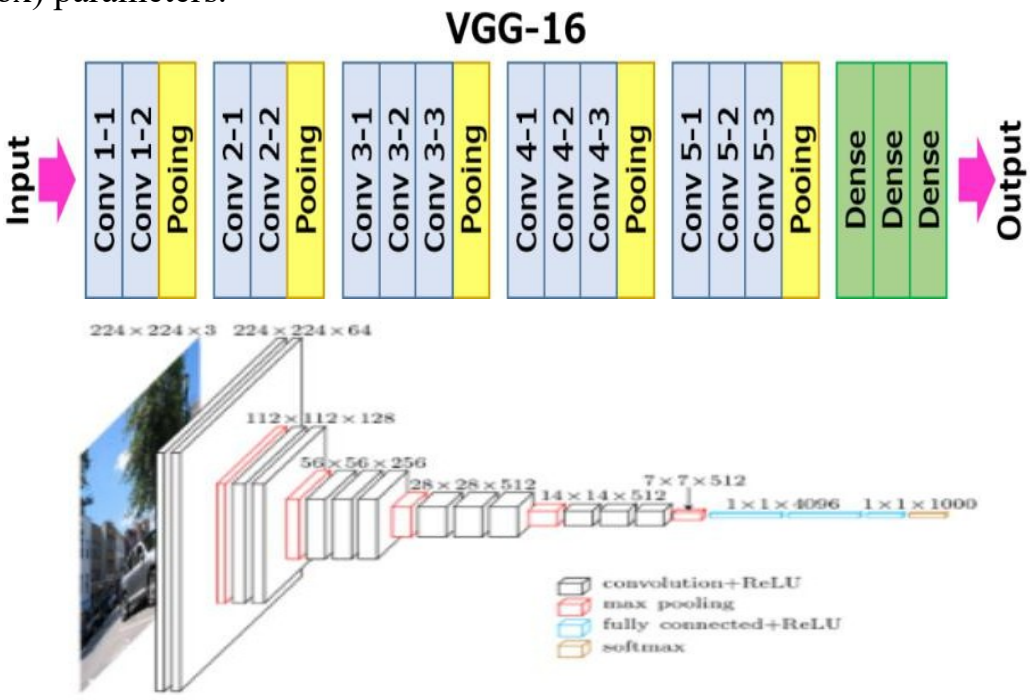
1. Train the entire model.

2. Train some layers and leave the others frozen.

3. Freeze the convolutional base. This case corresponds to an extreme situation of the train/freeze trade-off. The main idea is to keep the convolutional base in its original form and then use its outputs to feed the classifier. You're using the pre-trained model as a fixed feature extraction mechanism, which can be useful if you're short on computational power,your dataset is small, and/or pre-trained model solves a problem very similar to the one you want to solve. We use VGG16 Pre-trained model in our case.

6.6 **VGG16**

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.



Chapter 7

PROJECT PLANNING

Let's briefly discuss the data preparation and the implementation of this project on real-life car images using Mask R CNN.

7.1 DATA COLLECTION

Although we will leverage transfer learning from suitable pre-trained(weights) CNN architecture, we need to customize the network for our specific use to minimize application specific loss. So we will run train the n/w on use of images of car damages, collected from Google, out of those some images are used for train and are used for validation purpose.

Car Damage Images

7.2 DATA ANNOTATION

We need to label the data. In computer vision object detection or object localization context, this labeling is called annotation. Precisely for our application, it is identifying the region of damage in an image and marking them accurately along the boundary of the scratch.For annotation purpose, We used is the VGG Image Annotator(VIA) at this link.

7.3 ENVIRONMENT SET-UP

This is one of the important steps before training the model on collected images and annotations(labels), as We will use Mask R-CNN' to leverage a few pre-trained CNN n/w weight matrices built on different standard datasets like ImageNet etc. and custom functions such as, data processing and preparation, configuration setup, model training, creating log-file to save iteration wise weight-matrix objects & n/w losses, object detection, masking detected localized areas etc.

7.4 NETWORK TRAINING

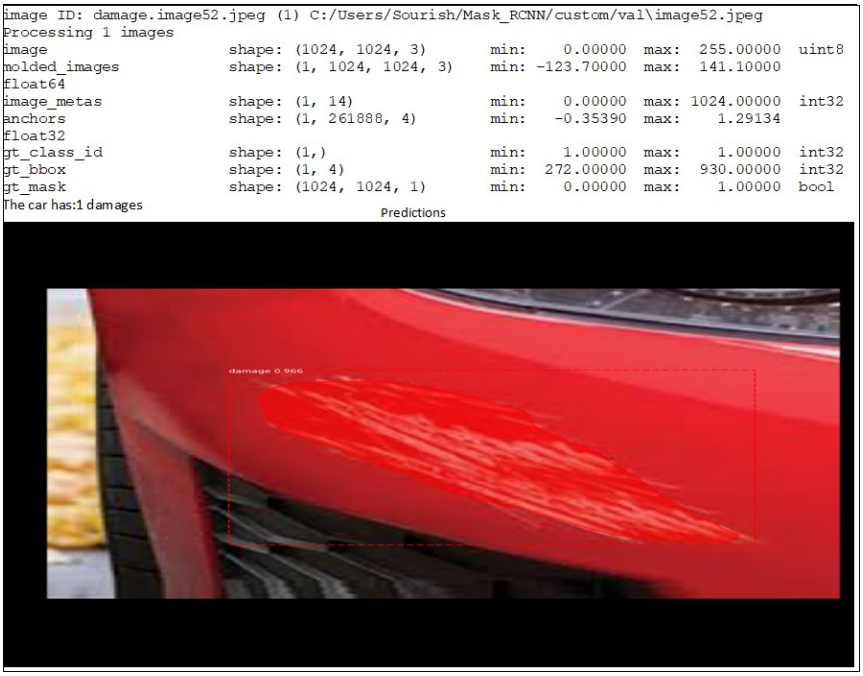
Now we need to refine the base with model training on real images and after each iteration(epoch) updated weight matrix. Also, iteration/epoch wise loss statistics are saved to monitor it in TensorBoard.

7.5 MODEL VALIDATION

As each iteration(epoch) wise updated weight matrix is saved in 'log'. Also, iteration/epoch wise loss statistics are saved to monitor it in Tensor Board.

7.6 MODEL PREDICTION

After satisfactory and desirable loss monitoring ideally monotonically decaying both training and validation loss, we can test the model object on randomly picked validation images to see the prediction (car damage masking) accuracy.



Chapter 8

IMPLEMENTATION

The implementation of this project involved the following steps:

- Import the necessary packages
- Import the required dataset and clean the dataset
- Split the dataset into a training and test set
- Apply algorithms: Logistic Regression Compare the accuracies on the training and test set.
- Generate graphs comparing train and test set accuracies for different algorithms

The implementation is described in detail below.

Model Implementation Mechanism

Model 1: Check Car or Not

Our first objective is to develop and train a model which can detect whether the given image uploaded by the user is an image of a car or not. First we import the necessary packages:

```
import numpy as np
import json
from keras.applications.vgg16 import VGG16
from keras.preprocessing.image import img_to_array, load_img from
keras.applications.imagenet_utils import preprocess_input
```

After importing necessary packages we uploaded the vgg16 pre-trained model:-

```
vgg16 = VGG16(weights='imagenet')
```

We then prepare the images and predicted using vgg.predict().

Imported the dataset as follows:

```
CLASS_INDEX_PATH='C:/Users/Ashar/Desktop/cdd/imagenet_class_index.json'
CLASS_INDEX = json.load(open(CLASS_INDEX_PATH))
```

Defined a function that categorizes different cars

get_car_categories()

Defined the final function to check whether the given image is an image of a car or not and print the desired results.

car_categories_gate()

As we can see,The desired result below:

First Check - car or not

```
In [23]: def car_categories_gate(image_path, cat_list):
             img = prepare_image(image_path)
             out = vgg16.predict(img)
             top = get_predictions(out, top=5)
             print ("Validating that this is a picture of your car...")
             for j in top[0]:
                 if j[0:2] in cat_list:
                     print (j[0:2])
                     return "Validation complete - proceed to damage evaluation"
             return "Are you sure this is a picture of your car? Please take another picture (try a different angle
         or lighting) and try again."

In [25]: car_categories_gate('C:/Users/Ashar/Desktop/image2.jpg', cat_list)

         Validating that this is a picture of your car...

Out[25]: 'Are you sure this is a picture of your car? Please take another picture (try a different angle or lightin
         g) and try again.'
```

Model 2: Check Car is Damaged or Not

Our second objective is to develop and train a model which can detect whether the given car image uploaded by the user is an image of a damaged car or not. First we import the necessary packages:

```
import os import glob
import datetime
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.preprocessing import image
from keras.models import Model
from sklearn.preprocessing import LabelEncoder
import numpy as np import
h5py
import json import time
```

With the help of our previous model developed ,we proceed further. We first filter the warnings,config the variables and we are loading the base VGG16 model with weights and then excluding the top dense layer.After succesfully loading we then encode the labels using LabelEncoder. We then save the features ,labels and then the models and weights.

We then use Logistic Regression to model the probability of a certain class or event existing such as car damaged/not.
Importing the necessary packages:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Now we import the features and labels and split the dataset into train and test datasets.Next we Created the model and evaluated the model of test data and then obtained the accuracy:

```
print ("[INFO] creating model...")
model = LogisticRegression(random_state=seed)
model.fit(trainData, trainLabels)
```

Obtained Accuracy is represented using Seaborn Heatmap.

After doing all these now it's time to obtain the result of car damaged or not. We again import the necassary packages and then load the trained Logistic Regression classifier.After this we load an image of a damaged car and got the below results:

```
In [10]: #read in the image, pre-proces the image and make predictions
         img = image.load_img('C:/Users/Ashar/Desktop/image10.jpg', target_size=image_size)
         x = image.img_to_array(img)
         x = np.expand_dims(x, axis=0)
         x = preprocess_input(x)
         feature = model.predict(x)
         flat = feature.flatten()
         flat = np.expand_dims(flat, axis=0)
         preds = classifier.predict(flat)

In [11]: preds
Out[11]: array([0], dtype=int64)

 In [8]: label_check1 = ['Car is Damaged', 'Car is not Damaged']

In [12]: label_check1[preds[0]]
Out[12]: 'Car is Damaged'

 In [7]: preds[0]
 Out[7]: 1
```

As we can see, The car is damaged.

Model 3: Check Car Damage Location

After building the above two models,time to develop a model that can check the location of where the damage has occurred.We again start with importing the necessary packages:

```
import os
import glob
import datetime
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.preprocessing import image
from keras.models import Model
from sklearn.preprocessing import LabelEncoder
import numpy as np
import h5py
import json
import time
```

With the help of our previous models developed ,we proceed further.We first filter the warnings,config the variables and we are loading the base VGG16 model with weights and then excluding the top dense layer.After succesfully loading we then encode the labels using LabelEncoder. We then save the features ,labels and then the models and weights. Next we then use Logistic Regression model the probability. Importing the necessary packages:

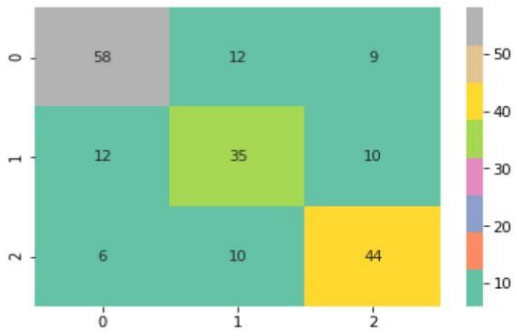
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Next we imported the features and labels and split the dataset into train and test datasets.Next we Created the model and evaluated the model of test data and then obtained the accuracy:

```
print ("[INFO] creating model...")
model = LogisticRegression(random_state=seed)
model.fit(trainData, trainLabels)
```

Obtained Accuracy is represented using Seaborn Heatmap.

```
In [13]: accuracy = ((58+35+44)/196)*100

In [14]: accuracy
Out[14]: 69.89795918367348

In [15]: sns.heatmap(cm,
                     annot=True,
                     cmap="Set2")
         plt.show()
```



Next it's time to obtain the result of car damage location.We again imported the necessary packages and then loaded the trained Logistic Regression Classifier.After this we loaded an image of a damaged car and got the below results:

```
In [13]: #read in the image, pre-proces the image and make predictions
         img = image.load_img('C:/Users/Ashar/Desktop/damaged car 3.jpg', target_size=image_size)
         x = image.img_to_array(img)
         x = np.expand_dims(x, axis=0)
         x = preprocess_input(x)
         feature = model.predict(x)
         flat = feature.flatten()
         flat = np.expand_dims(flat, axis=0)
         preds = classifier.predict(flat)

In [14]: preds
Out[14]: array([2], dtype=int64)

In [11]: label_check1 = ['Front Damage', 'Rear Damage', 'Side Damage']

In [15]: label_check1[preds[0]]
Out[15]: 'Side Damage'
```

Out of the 3 location labels of damage I.e Front,Rear,Side, It shows that the location of the damage is 'Side Damage'.

Model 4: Check Car Damage Severity

After building the above three models,it's time to develop a model that can check the severity of the damage that has occurred.We again start with importing the necessary packages:

```
import os
import glob
import datetime
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.preprocessing import image
from keras.models import Model
from sklearn.preprocessing import LabelEncoder
import numpy as np
import h5py
import json
import time
```

With the help of our previous models developed ,we proceed further. We first proceed with filtering the warnings, config the variables followed by loading the base VGG16 model with weights and then excluding the top dense layer.After successfully loading ,we encode the labels using LabelEncoder. We then save the features ,labels and then the models and weights. Next we then use Logistic Regression to model the probability. Importing the necessary packages:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Now we import the features and labels and split the dataset into train and test datasets.Next we Created the model and evaluated the model of test data and then obtained the accuracy:

```
print ("[INFO] creating model...")
model = LogisticRegression(random_state=seed)
model.fit(trainData, trainLabels)
```

Obtained Accuracy is represented using Seaborn Heatmap.

```
In [13]: accuracy = ((30+34+67)/196)*100

In [14]: accuracy
Out[14]: 66.83673469387756

In [15]: sns.heatmap(cm,
                     annot=True,
                     cmap="Set2")
         plt.show()
```

After doing all these now it's time to obtain the result of car damage severity. We again import the necassary packages and then load the trained Logistic Regression classifier. After this we load an image of a damaged car and got the below results:

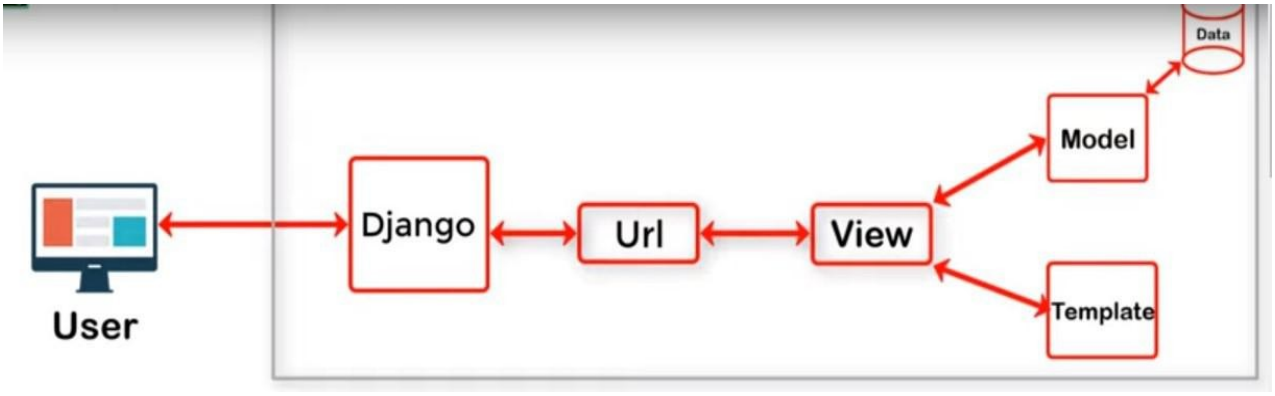
```
In [8]: #read in the image, pre-proces the image and make predictions
        img = image.load_img('C:/Users/Ashar/Desktop/damaged car 3.jpg', target_size=image_size)
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        x = preprocess_input(x)
        feature = model.predict(x)
        flat = feature.flatten()
        flat = np.expand_dims(flat, axis=0)
        preds = classifier.predict(flat)

In [9]: preds
Out[9]: array([1], dtype=int64)

In [6]: label_check1 = ['Minor Damage', 'Moderate Damage', 'Severe Damage']

In [10]: label_check1[preds[0]]
Out[10]: 'Moderate Damage'
```

Out of the 3 location labels of damage I.e Minor,Moderate,Severity Damage, It shows that the severity of the damage is 'Moderate Damage'.

DEPLOYMENT PHASE

Framework Used: DJango

Django is full of shortcuts to make Web developers' lives easier, but all those tools are of no use if you can't easily deploy those in your sites. Since Django's inception, ease of deployment has been a major goal.



There are many options for deploying Django application, based on the architecture or particular business needs.Django, being a web framework needs a web server in order to operate. And since most web servers don't natively speak Python, we need an interface to make that communication happen.

Django supports two interfaces: WSGI and ASGI.

WSGI is the main Python standard for communicating between Web servers and applications, but it only supports synchronous code.ASGI is the new, asynchronous-friendly standard that will allow your Django site to use asynchronous Python features, and asynchronous Django features as they are developed.We also considered how tol handle static files for the application, and how to handle error reporting.

The application object

The key concept of deploying with WSGI is the application callable which the application server uses to communicate with your code. It's commonly provided as an object named application in a Python module accessible to the server.

The start project command creates a file <CarApp>/wsgi.py that contains such an application callable.It's used both by Django's development server and in production WSGI deployments.

WSGI servers obtain the path to the application callable from their configuration. Django's built-in server, namely the run server command, reads it from the WSGI_APPLICATION setting. By default, it's set to <CarApp>.wsgi.application, which points to the application callable in <CarApp>/wsgi.py.

Configuring the settings module:-

When the WSGI server loads your application, Django needs to import the settings module — that's where your entire application is defined.

Django uses the DJANGO_SETTINGS_MODULE environment variable to locate the appropriate settings module. It must contain the dotted path to the settings module. You can use a different value for development and production; it all depends on how you organize your settings.

If this variable isn't set, the default wsgi.py sets it to mysite.settings, where mysite is the name of your project. That's how runserver discovers the default settings file by default.

Applying WSGI middleware:-

To apply WSGI middleware you can wrap the application object. For instance you could add these lines at the bottom of views.py:

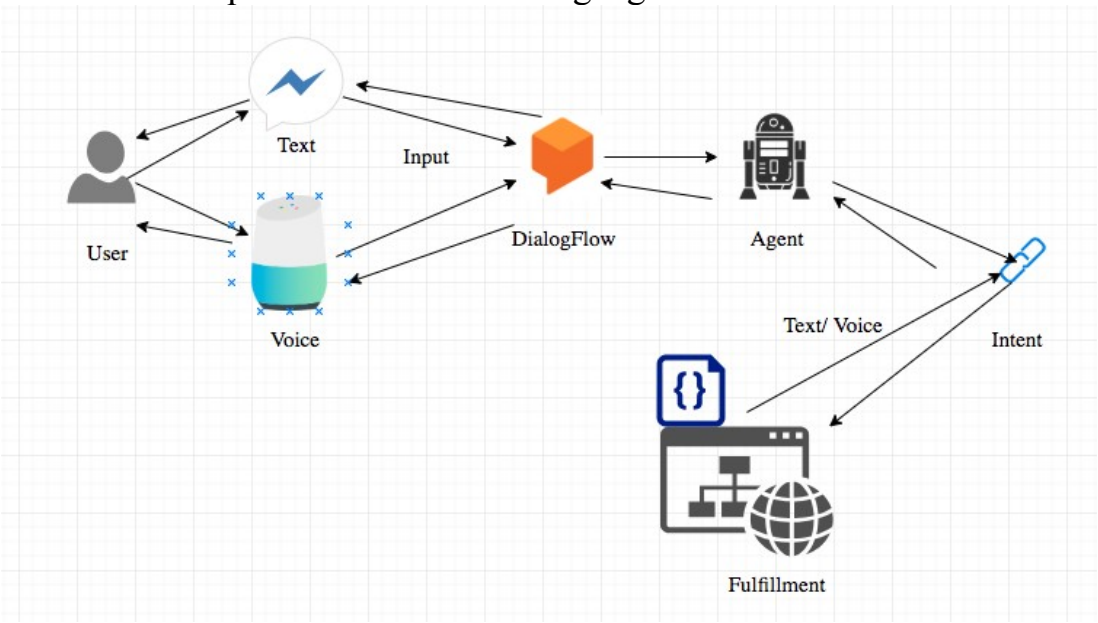
```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.core.urlresolvers import reverse
from django.urls import reverse
from carapp.models import PicUpload
from carapp.forms import ImageForm
```

Whenever we visit a website, Data is what we want, along with a layout. MVT(MODEL, VIEW, TEMPLATE)is a software design pattern that binds these two together. It is used for displaying dynamic content. The data, the layout and the logic is separated from one another MVT . Here, the MODEL is used to fetch the data, and the TEMPLATE is concerned with the html files(the designing of the layout).These two entities are linked via VIEW. VIEW is concerned with the logic and functionalities of the website. So, whenever a user sends a request, the django framework will direct it to the urls.py file, which has all the mappings. That navigation will be sent to VIEWS, which also binds the data and the layout together. The data that is to be displayed in the templates will be decided by the VIEWS, through which the connection with the MODELS is also done.

CHAT-BOT DEVELOPMENT PHASE

Chatbot is a program that can conduct an intelligent conversation. It should be able to convincingly simulate a human behaviour and pass the turing test.

Tool Used : Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant (bySpeaktoit), a virtual buddy for Android, iOS, and Windows Phone smartphones that performs tasks and answers users' question in a natural language.



Flow of conversation within DialogFlow

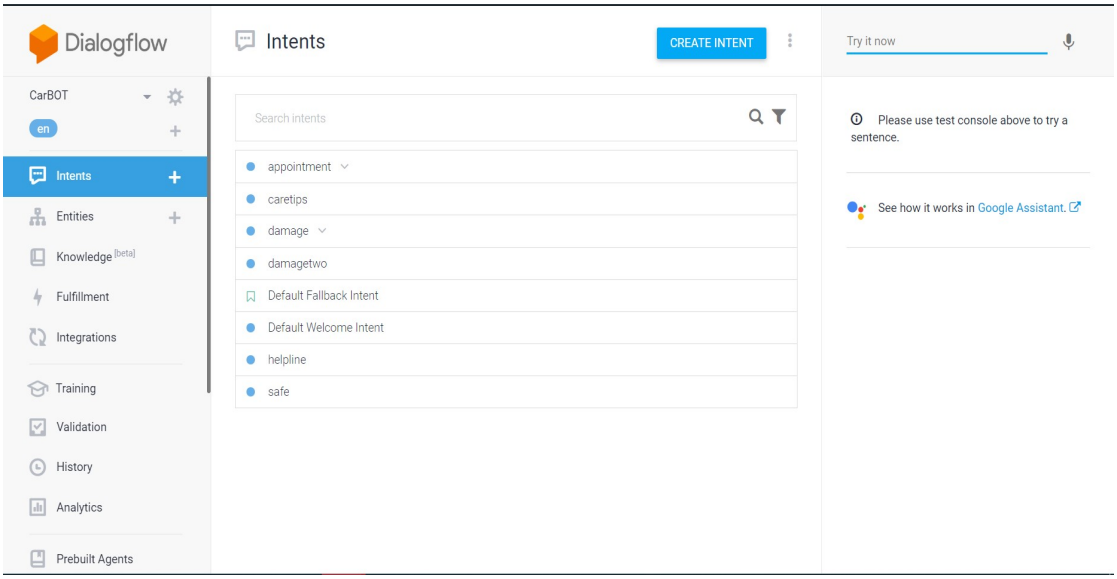
Setting up Dialogflow account.:-

Let's jump into starting and setting up the environment where we created our bot!.

- Goto : https://dialogflow.com/
- Create an account with a gmail account, and "agree" to the terms & conditions.

Creating Intents:-

Intent: Support or the service that the user wants from the agent. Intent is configured by the developers. Intent determines the action by the code. We created the following intents where each specific intent is responsible for performing a certain task.After successfully creating the intents we train each intent by providing some sample training phases example that user may ask possibly and then integrated our chatbot with our website.



Intents created

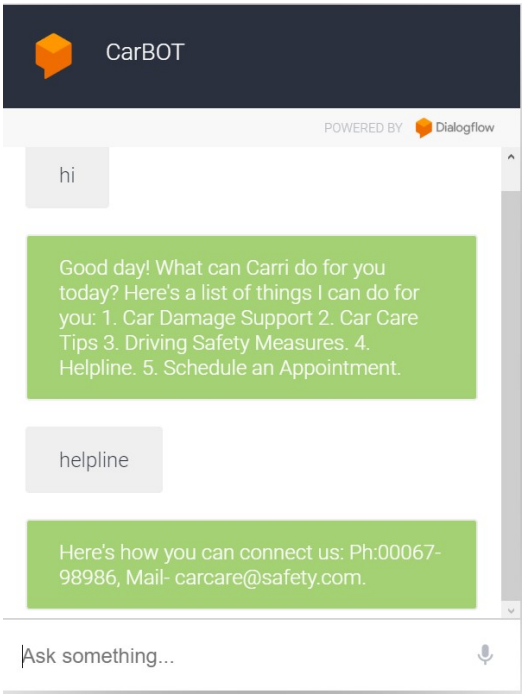
Integrating the Chatbot in the Car portal:-

Dialogflow Messenger brings a rich UI for Dialogflow that enables developers to easily add conversational agents to websites.

We added this agent chatbot to our website by the following code below:

```
<script src="https://www.gstatic.com/dialogflow-
console/fast/messenger/bootstrap.js?v=1"></script>
<df-messenger
  intent="WELCOME"
  chat-title="CarBOT"
  agent-id="92d5343b-0c89-4c92-b4c6-9ecc2482780f"
  language-code="en"
></df-messenger>
```

Here's the result:-



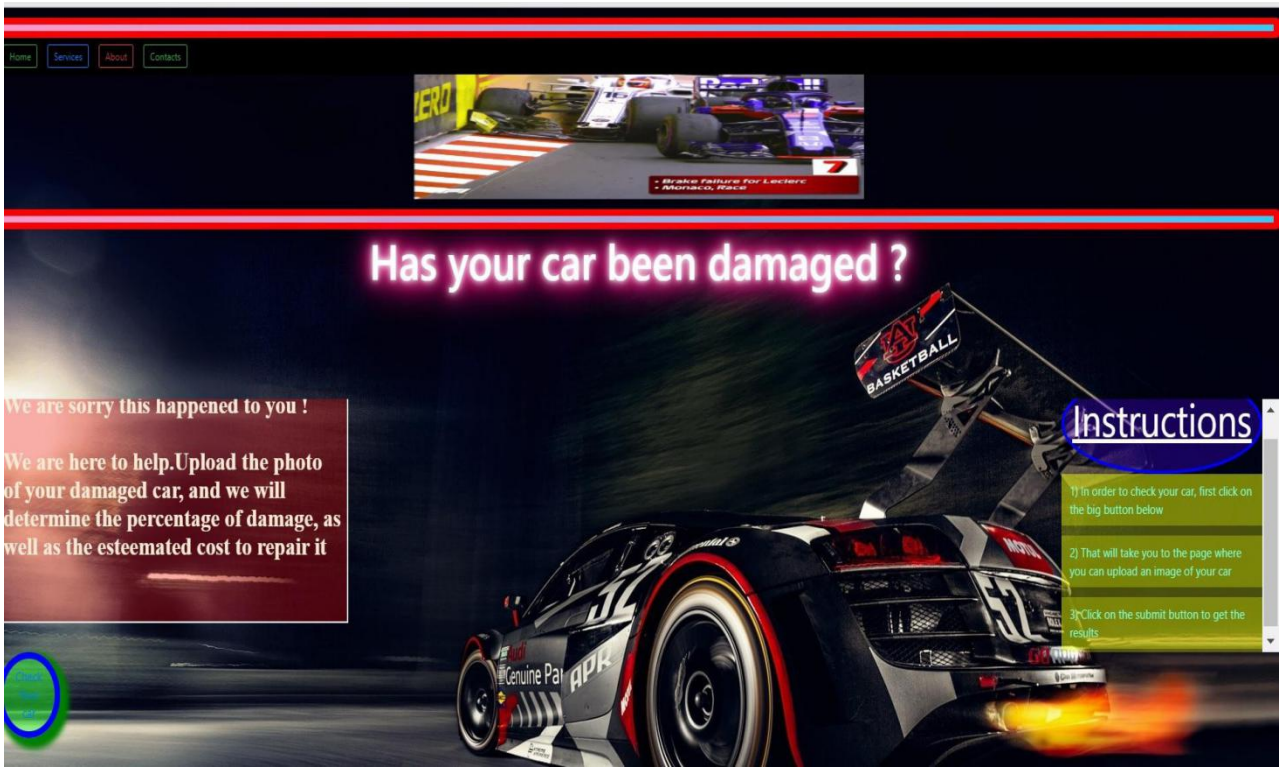
Chapter 9

SCREEN SHOTS OF PROJECT

9.1 HOMEPAGE

Options Available-

1. Log-in/Sign-up.
2. Read Instructions.
3. Click on Check Car.



9.2 SERVICE PAGE

Options Available-

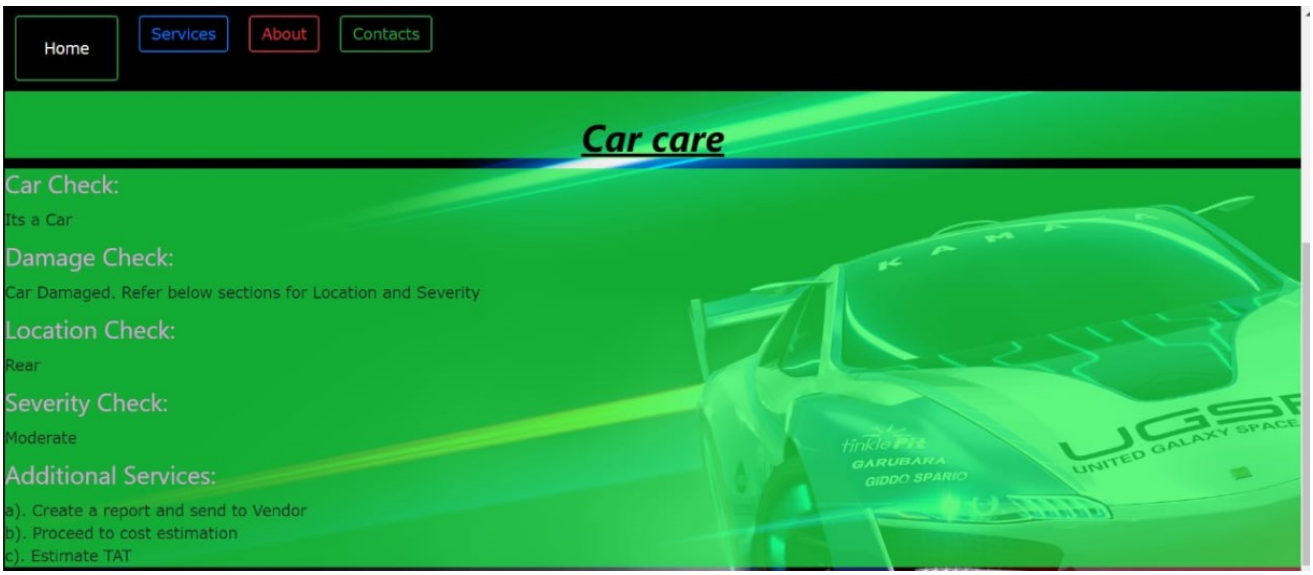
1. Click Choose File/Image.
2. Select/Upload Image.
3. Submit.

9.2 RESULTS PAGE

Options Available-

1. Predicted the Image is of a Car/Not Car.
   Car is Damaged or not.

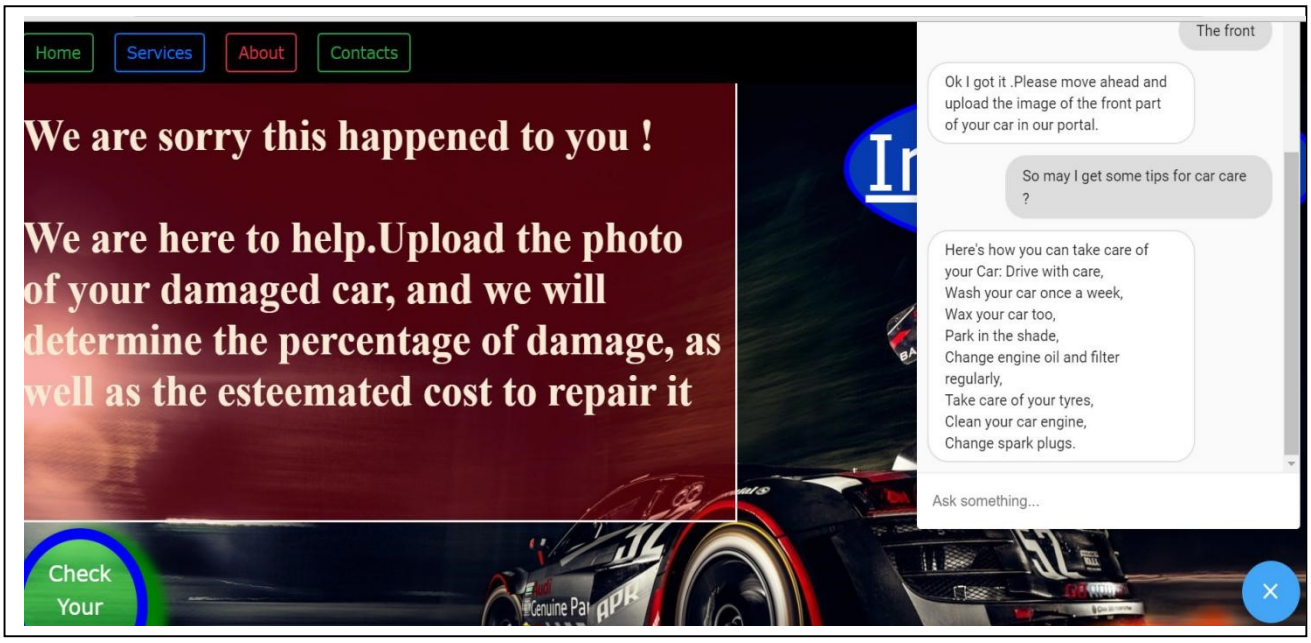
2. Location of the Damage.
3. Severity of Damage.
4. Additional Services.



9.4 **HOMEPAGE**-(Chatbot Support)

Options Available-

1. Car Damage Check procedure info.
2. Helpline.
3. Car Care Tips.
4. Car Driving Safety Tips
5. Schedule an Appointment.



Chapter 10

Conclusion and Future Scope

10.1 Conclusion

In this paper, we proposed a deep learning based solution for car damage classification. Since there was no publicly available dataset, we created a new dataset by collecting images from web and manually annotating them. We experimented with multiple deep learning based techniques such as training CNNs from random initialization, Convolution Autoencoder based pre-training followed by supervised finetuning and transfer learning. We observed that the transfer learning performed the best. We also note that only car specific features may not be effective for damage classification. It thus underlines the superiority of feature representation learned from the large training set.

10.2 Future Scope

Used car dealers/car insurance company can install infrastructure with high- resolution cameras at suitable angles and locations to click standardized(size) images of different car body sections(front, back, side etc.) and can detect all possible exterior damages in cars. This concept can be used as a mobile app as an API solution, which can ease the car evaluation process.

Further after detection and masking of car damage, the process can help the car evaluators/claim settlement personnel in quantifying the damage severity, in terms of dimensions and approximate relative area(w.r.t. the car surface area) of damage. Most importantly since we're leveraging transfer learning, we don't have to collect many images and subsequent annotation and as the model training starts from trained weights('coco'), we don't need to train too long. Also, this concept can be extended to detect other types of visible car damages/faults as well.

References

1)S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.

2)K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

3)Tammina, Srikanth. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications (IJSRP). 9. p9420. 10.29322/IJSRP.9.10.2019.p9420.

4) Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

5)M. Shaha and M. Pawar, "Transfer Learning for Image Classification," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp. 656-660, doi: 10.1109/ICECA.2018.8474802.

6)https://docs.djangoproject.com/en/3.0/

7) https://github.com/nicolasmetallo/car-damage-detector

8) https://www.kaggle.com/anujms/car-damage-detection

9)https://www.researchgate.net/publication/322671990_Deep_Learning_Based_Car_Damage_Classification

INDIVIDUAL REPORT

CAR DAMAGE IDENTIFICATION MODEL

CHIRAYATA CHATTERJI
1705779

Abstract: The aim of the project is to build a custom Mask R-CNN 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 they can assess damage from them. This model can also be used by lenders if they are underweitting a car loan especially for a used car**.**

Individual contribution and findings: It has been a pleasure working on this project as I have made a significant contribution to it. My role was to deploy the machine learning model on the web server, as well as the front end part(styling). Since the entire code is written in python and its libraries, I chose django as the framework. I have made 3 pages in the website. The welcome page, the page for uploading the car image, and the result page demonstrating damages(if any). I have first made a seperate app named 'carapp', under the main project folder.The 'views.py' of this app is where the functionality of the website lies. The machine learning codes have been added here under different functions in an organised way. The important modules and packages have also been imported here. The three main functions navigate to the 3 different pages of the website.The corresponding pages(urls) are mapped to the 'urls.py' of the main 'cardamage' folder. The MVT(Model View Template) concept of django brings an amazing flow in the entire process. Next comes the styling part of the website. I have created the web pages using html,css and frontend javascript. I have kept all these files under the 'templates' folder. The images and gifs used have been kept in the 'static' folder.Another task I had to perform is merging the dynamic content generated from the machine learning code to the static html files. For that I have also used DTL(django template language). I have also implemented the chatbot in the home page, which was made by one of my teammates. As far as the styling part is concerned, I have used bootstrap framework for css. The dropdowns, the navbars, the frames all I tried to do as good looking as possible.The different elements are separated via classes, ids, as well as putting some of them under the div and container tags.

The bootstrap cdn and other important links have been added at the very beginning. Next comes the css code under the style tags. The text fonts, the background colors(in the rgb format), the margin, padding , border properties for specific elements have been included here.The frontend javascript code has been added under the script tags at the bottom, for example, to create the glowing effect of the huge texts at the first and last pages. In short, it was a wonderful journey,and I am quite satisfied with my own performance in my area which had its own role in leading the project to success.

Individual contribution to project report preparation: Since I deployed the project in my local host server, I have made my contribution in the implementation part. There, I have mentioned the details about how I put the machine learning model in an organised way, as mentioned earlier, the MVT concept by which I mapped the pages to the urls.py file through the functions of the views.py, how I merged the dynamic outputs of the code with the static html files using DTL, how I designed the web pages using html, css and front end javascript, and so on. I also sent the screenshots of the web pages for the main project report.

Individual contribution for project presentation and demonstration: I have created the web pages after deploying the model to the server, and so, it is be a part of my presentation. I have demonstrated how the website works, as well as the designing and styling of the web pages. I have also demonstrated the codes in the different files, as well as the work flow of the django framework.

Full Signature of Supervisor:
…….

Full signature of the student:
……..

INDIVIDUAL REPORT

CAR DAMAGE IDENTIFICATION MODEL

PRIYOTOSH SAHA
1705869

Abstract: This project aims to build a custom Mask R-CNN 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 they can assess damage from them. This model can also be used by lenders if they are underwriting a car loan especially for a used car.

Individual contribution and findings: The project was beneficial and I have learned a lot about Convolution Neural Network model creation while working on this project would not have been possible without the help and support of many individuals and the organization.I have given the idea of the whole project and my role was to implement the third checking part of the prediction model. The whole integrated model was implemented by me. According to the project development final report,

i) I have implemented the part of checking the exact damage location(Front, Rear, Side) for the provided car images of the test dataset. I have imported features and label encoders using the VGG16 model. After that I have implemented logistic regression to train the model. The path setting of the folders of labels and features was the most challenging thing during the model implementation. It has three categories of labels i.e- front, rear, and side. After creating the config file I have loaded the base model VGG16 to process the images(224,224) and excluded the top dense layer. After setting all of the parameters and preprocessing images encoded the labels.After trained the whole model I have saved the file in h5py format for the future deployment process. I have used the train test splitting and confusion matrix to fit the model. I have saved the classifier model in a pickle file. The accuracy of that model was around 70%.

ii) My other contribution was to integrate all of the four developed models made by the group individuals. It was the integrated part that had to deploy in web APIs through the Django frame. First I have loaded all of the four models based on VGG16 and pickle file to their path for checking purpose.Then I have executed every model accordingly in that integrated part by creating functions. Finally I have finished the model combination and created a function named engine that can take the input images and can check all of the criteria to detect a damaged car.

Individual contribution to project report preparation:

I am very much satisfied with my contribution to the final project report. I have contributed a little bit to the final group project report by referring to the IEE format and another kind of research paper. I have studied so many papers to upgrade the project report quality. I have provided the DFD and all of the required use case diagrams to enhance the features of our final report.

Individual contribution for project presentation and demonstration:
According to my contribution towards the project, I would like to demonstrate the part of the third checking process of the car damage detection and I would also like to present how to implement the integration part of the whole models using Django framework.

Full Signature of Supervisor: ……

Full signature of the student: ……..

INDIVIDUAL REPORT

CAR DAMAGE IDENTIFICATION MODEL

Shubham Banerjee
1705825

Abstract: This project aims to build a custom Mask R-CNN 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 they can assess damage from them. This model can also be used by lenders if they are underwriting a car loan especially for a used car.

Individual contribution and findings: It had been a great learning curve for me as I learnt the principle concepts of Computer Vision which involves getting insights and perform analysis on Image Dataset.I have been fortunate to contribute to the following parts of the project:

i) Since machine learning models works on training these models with certain feature Datasets and its labels and then it can give its predictive analysis on unknown data,I had to acquire lot of images consisting of damaged or Undamaged cars or any random image to better train the model so that it can achieve greater accuracy in predicting unknown image data.Then I had to run preprocessing steps in the training data like removing unnecessary noise from images,reshaping the images to a fixed size,producing more images from limited images with the technique of Data Augmentation which involved position,augmentation(Scaling,Cropping,Flipping,Padding,Rotation,Translation,Transformation) and Color Augmentation(Brightness,Contrast,Saturation,Hue).

ii) Then our first objective while training was to determine whether a image is car or not.For this I used the technique of transfer learning which uses pre-trained model which had been trained on a lot of images.One of the popular pre-trained model was VGG16 which consisted a lot of convolution layers,the latter is used in convolving the images which the computer sees as an array of pixel values ranging from 0-255 and having a certain resolution,with a weighted filter which helps in extracting meaningful insights from a image like detecting edges,detecting shapes etc.I also had to exclude the VGG16's dense layer and attached with my own defined dense layer having softmax activation which gave the probability of a presence of a car or not. During the learning phase,these filters are initialized with random weights and biases and after validation on a single step or epoch,the error is backpropagated to tune the weights and the steps are repeated until certain level of accuracy is reached.Hence the model was successfully trained to detect car images.

iii) Finally as part of the fourth stage ,I had to determine severity of the Car Damage .With the help of the previous models developed,I first filtered the warnings,config the variables and then loading the base VGG16 model with pre-trained weights and biases and excluding the to dense layer .After successfully loading we then encode the labels using LabelEncoder.Finally we use the logistic regression to model the probability of the damage caused.

Individual contribution to project report preparation:

I have been fortunate to contribute in preparing the report by explaining the Fundamental processing steps required to work on images by explaining convolutional neural networks,transfer learning,various pre-trained models like ResNet,VGG16,Inception etc. I also helped in the practical implementation of the project.

Individual contribution for project presentation and demonstration:

In this regard I contributed in explaining and demonstrating the business and real world applications of our project.I have also demonstrated the that what's our main aim behind developing this project and what we are trying to achieve with this.

Full Signature of Supervisor:
…….

Full signature of the student:
……..

INDIVIDUAL CONTRIBUTION REPORT

CAR DAMAGE IDENTIFICATION MODEL

SWAYAMDIPTA SAHA
1705837

Abstract: The aim of this project is to build a custom Mask R-CNN 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 they can assess damage from them. This model can also be used by lenders if they are underwriting a car loan especially for a used car.

Individual contribution and findings: It has been a great journey,developing this amazing project and I have been fortunate enough to contribute to two of the most integral parts of this project which are as follows:

i. My first task was to develop and train a model which can detect whether the given car image uploaded by the user is an image of a damaged car or not.With the help of our previous models developed, I proceeded further. I first filtered the warnings,configured the variables and loaded the base VGG16 model with weights and then excluded the top dense layer.After successfully loading then I encoded the labels using Label Encoder,then saved the features ,labels and then the models and weights.I applied Logistic Regression machine learning algorithm to model the probability of a certain class or event existing such as car is damaged/not.Then I imported the features and labels and split the data-set into train and test data-sets.Next, I Created the model and evaluated the model of test data and then obtained the accuracy which came out to be 89.94%.After importing the necessary packages and loading the trained Logistic Regression classifier,I tested and loaded a sample image of a damaged car and got the required results.My model was predicting correctly with a good accuracy.

ii. To build a chatbot using Dialogflow.Chatbot is a program that can conduct an intelligent conversation. It should be able to convincingly simulate a human behaviour and pass the turing test. Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations.I used dialogflow to develop the chat-bot application and successfully integrated the chat-bot with our Car Damage web portal.The main purpose of building the chat-bot is to provide the user a better UI and interactive web page.This chat-bots contains some basic functionalities like Car care tips,Car Damage Support,Helpline number,Car driving safety measures etc.

Individual contribution to project report preparation: I have been fortunate enough to contribute to two of the most integral parts of this project i.e,

i. In the Implementation phase of the report I have given the detailed description of my model that I had developed to classify image whether car is damaged or not .I have presented proper screen-shots of the packages I used in developing the model along with the snapshots of obtained Accuracy which I represented using Seaborn Heatmap.At the end I have shared the screenshot of the model predicting the image result.ii. My second contribution is representing the Architectural View Decomposition Procedural Design/Diagram.I have designed and represented the Use-Case Diagram and Activity Diagram of our project.This diagrams represents how a user would interact with our web portal and what are the funtionalities available to the user and their flow.

ii. My third contribution is representing the way Dialogflow is used in this project in building the chat-bot successfully.Dialogflow (formerly Api.ai, Speaktoit) is a Google owned developer of human–computer interaction technologies based on natural language conversations.I mentioned the mechanism of how I used dialogflow to develop the chat-bot application and successfully integrated the chat-bot with our Car Damage web portal.

Individual contribution for project presentation and demonstration:

Coming to the presentation and demonstration part, I have contributed in preparing the introduction,objective and the abstract behind developing this project.I have also contributed in designing the presentation in terms of styling and customization. I have demonstrated the same to our project evaluator.I have demonstrated, that what's our main aim behind developing this project and what we are trying to achieve with this.I have also demonstrated the part I have contributed in this project as mentioned earlier i.e Car Image Damage Check model and my chatbot application which I have built using Dialogflow successfully.

Full Signature of Supervisor: Full signature of the student:

……. ……..

INDIVIDUAL REPORT

CAR DAMAGE IDENTIFICATION MODEL

RISHIT DUBEY

1705768

Abstract: The aim of the project is to build a custom Mask R-CNN 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 they can assess damage from them. This model can also be used by lenders if they are underweitting a car loan especially for a used car**.**

Individual contribution and findings: It has been a pleasure working on this project as I have made a significant contribution to it. My role was to do the research on the project enhancement and further scope for its significance contribution in near future. The future scope of this project can be quite relevant. Other than a car's functionality, its body external damages like scratches, dents etc has a vital role to decide accurate pricing of the vehicle. This concept will also help car insurers in assessing the damage automatically and in processing claim faster. In future, this project can detect the damage location faster and help pricers to quantify the damage without doing it manually, helping them to save time and energy. Other future scope is that we can add some modifications like as a person uploads the image of the car's damaged area, he/she gets a estimate about the costing and the nearest service centre with the help of location and gps.

Individual contribution to project report preparation: Since I did some research in this project, the data-set or the data collection part has been carried out . This data-set helps to find the damaged area on the car's external body. This dataset has been collected from google i.e kaggle and some images are used for train and remaining for the validation purpose. With help of this dataset, it helps to detect different types of car damage. The number of images in the input data is small relative to the number of classes and has a significant amount of variation. The image counters with the dataset to search for the exact damaged location on the car.

Individual contribution for project presentation and demonstration: I have comtributed with the research on the datasets and future scope for this project.

Full Signature of Supervisor: Full signature of the student:

……. ……..

Plagiarism Checker X Originality Report

Plagiarism Quantity: 25% Duplicate

Date	Friday, June 05, 2020
Words	2156 Plagiarized Words / Total 8474 Words
Sources	More than 78 Sources Identified.
Remarks	Medium Plagiarism Detected - Your Document needs Selective Improvement.

A PROJECT REPORT on ﷺCAR DAMAGE IDENTIFICATION MODELﷺ Submitted to KIIT Deemed to be University In Partial Fulfilment of the Requirement for the Award of BACHELORﷺS DEGREE IN COMPUTER SCIENCE AND ENGINEERING BY CHIRAYATA CHATERJI 1705779 PRIYOTOSH SAHA 1705869 RISHIT DUBEY 1705768 SHUBHAM BANERJEE 1705825 SWAYAMDIPTA SAHA 1705837 UNDER THE GUIDANCE OF PROF. BISWAJIT SAHOO SCHOOL OF COMPUTER ENGINEERING KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY BHUBANESWAR, ODISHA - 751024 May 2020 A PROJECT REPORT on ﷺCAR DAMAGE IDENTIFICATION MODELﷺ Submitted to KIIT Deemed to be University In Partial Fulfillment of the Requirement for the Award of BACHELORﷺS DEGREE IN COMPUTER SC. AND ENGINEERING BY CHIRAYATA CHATTERJI 1705779 PRIYOTOSH SAHA 1705869 RISHIT DUBEY 1705768 SHUBHAM BANERJEE 1705825 SWAYAMDIPTA SAHA 1705837 UNDER THE GUIDANCE OF PROF.

BISWAJIT SAHOO SCHOOL OF COMPUTER ENGINEERING KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY BHUBANESWAR, ODISHA -751024 May 2020 KIIT Deemed to be University School of Computer Engineering Bhubaneswar, ODISHA 751024 CERTIFICATE This is certify that the project entitled ﷺCAR DAMAGE IDENTIFICATION MODELﷺ submitted by CHIRAYATA CHATTERJI 1705779 PRIYOTOSH SAHA 1705869 RISHIT DUBEY 1705768 SHUBHAM BANERJEE 1705825 SWAYAMDIPTA SAHA 1705837 is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci- ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance. Date: / / (Prof. Biswajit Sahoo) Project Guide Acknowledgement We are profoundly grateful to Prof. BISWAJIT

Sources found:

Click on the highlighted sentence to see sources.

Internet Pages

<1% http://www.handbook.mq.edu.au/2019/Degre
<1% https://cis-india.org/telecom/telecom-ma
1% https://beta.vu.nl/nl/Images/stageversla
<1% https://www.semanticscholar.org/paper/Au
<1% https://www.kitces.com/blog/pseudodeduct
<1% https://jewjewjew.com/famous-games-socia
4% https://www.analyticsvidhya.com/blog/201
<1% https://www.scribd.com/document/49152131
<1% http://www.ddegjust.ac.in/studymaterial/
1% https://www.researchgate.net/profile/Man
1% https://www.ee.iitb.ac.in/student/~kalpe
<1% https://www.researchgate.net/publication
<1% https://www.nature.com/articles/s42256-0
<1% https://www.researchgate.net/publication
<1% https://cppsecrets.com/users/10019897108
<1% https://github.com/DjangoPeng/keras
<1% https://howto.lintel.in/python-matplotli
<1% https://towardsdatascience.com/data-visu
<1% https://www.geeksforgeeks.org/project-sc
<1% http://www.personal.psu.edu/jxx1/teachin