Go, change the world

RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi

Report
On

Software Engineering
20MCA21

Image classification using TinyML

Submitted in Partial Fulfillment of the Requirement
for the II Semester MCA

MASTER OF COMPUTER APPLICATIONS

By

1RV21MC071	**PIKU MAITY**
1RV21MC081	**RAKSHITH R**
1RV21MC083	**RANJITH KUMAR J**
1RV21MC114	**VAIBHAV KULKARNI**

Under the Incharge
of

Prof. Dr. S.S.Nagamuthu.Krishnan

Department of Master of Computer Applications
RV College of Engineering®, Mysuru Road
RV Vidyanikethan Post, Bengaluru – 560059

October -2022

Assignment Mark

ASSIGNMENT -I

SL NO		***MAX MARKS***	***MARKS***
1			
2			
3			

ASSIGNMENT -II

SL NO		***MAX MARKS***	***MARKS***
1			
2			
3			

DEPARTMENT OF

MASTER OF COMPUTER APPLICATIONS

Bengaluru– 560059

CERTIFICATE

This is to certify that Mr **Piku Maity**, USN **1RV21MC071**, Mr **Rakshith R**, USN **1RV21MC081**, Mr **Ranjith Kumar J**, USN **1RV21MC083** and Mr **Vaibhav Kulkarni**, USN **1RV21MC114** of 2nd semester Master of Computer Applications program has satisfactorily completed the Assignment titled **Image Classification using TinyML** in **Software Engineering - 20MCA21** as a part of Continuous Internal Assessment.

Dr. S.S.Nagamuthu.Krishnan	**Dr. Andhe Dharani**
Associate Professor	Professor and Director
Department of MCA	Department of MCA
RVCE, Bengaluru –59	RVCE, Bengaluru–59

Table of Contents

1. Software Requirements and Specifications

1.1 Introduction

Object Detection is the process of finding and recognizing real-world object instances such as car, bike, TV, flowers, and humans out of an images or videos. An object detection technique lets you understand the details of an image or a video as it allows for the recognition, localization, and detection of multiple objects within an image.

It is usually utilized in applications like image retrieval, security, surveillance, and advanced driver assistance systems (ADAS). Object Detection is done through many ways:

- Tensor Flow
- Feature Based Object Detection
- Viola Jones Object Detection
- SVM Classifications with HOG Features
- Deep Learning Object Detection

Humans can easily detect and identify objects present in an image. The human visual system is fast and accurate and can perform complex tasks like identifying multiple objects with little conscious thought. With the availability of large amounts of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy.

1.1.1 Purpose

The purpose of this document is to build a machine learning model to detect an object. This document comprises of the problem statement, requirements, functional description, etc. necessary to explain the functionality of the software being built. The focus here is to train the Machine Learning model over the existing dataset and then deploy the same model over current dataset collected and detect the object based on the input values received from the user through a GUI. This document covers the overall scope of coming up with solution to obtaining an effective Machine learning model that is to be implemented.

1.1.2 Document Conventions

Font: Times New Roman, Style: Bold, Size: 20	-	Page Headings
Font: Times New Roman, Style: Bold, Size: 18	-	Headings
Font: Times New Roman, Style: Bold, Size: 16	-	Subheadings
Font: Times New Roman, Style: Bold, Size: 14	-	Content (Subheading)
Font: Times New Roman, Style: - , Size : 14	-	Content(Body)
Square Bracket Enclosed Serial no. & linked	-	References

1.1.3 Intended Audience and Reading Suggestions

The document is intended for common users, requirements engineer, domain expert, developer or project manager. It is highly recommended to read the Introduction section to get an overview of the task and problem statement.

1.1.4 Product Scope

The software system with integrated webcam or an external webcam detects the object when an object is passed over it. The System then shall recognize the object as per trained model. The entered data can be just passed through same model to detection.

1.1.6 Definitions, Acronyms and Abbreviations

1.1.6.1 SVM

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

1.1.6.2 HOG

Histogram of Oriented Gradients, also known as HOG, is a feature descriptor like the Canny Edge Detector, SIFT. It is used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in the localized portion of an image.

1.1.6.3 SIFT

Scale Invariant Feature Transform (SIFT) is an image descriptor for image-based matching and recognition developed by David Lowe. This descriptor as well as related image descriptors are used for a large number of purposes in computer vision related to point matching between different views of a 3-D scene and view-based object recognition.

1.2 Overall Description

1.2.1 Product Perspective

This document is in context of the GUI which will implement the Machine Learning algorithm to detect the object. The model is meant to detect the object and recognize the object which is going to help in various domains like Agriculture, Traffic Monitoring, Criminal Identification etc. This model is going to detect the object depending upon trained data set.

1.2.2 Product Functions

- Take Inputs: Takes images as input for training the model
- Provide Output: Detects the object and recognizes the object, then displays the image name

1.2.3 Functional Requirements

- Able to store trained model onto the web session of the client without overloading the space
- Able to detect and recognize the object based on trained data set
- Do not authenticate general users to change the already trained model which is to be loaded

1.2.4 User classes and Characteristics

- This model is intended to be used by anyone, i.e. general user group
- The user group comprises of People, Organizations, Industries, Healthcare sector etc to detect the objects like Person, Things, Animals, Diseases etc
  All the above-mentioned users belong to same class and they all have same set of characteristics and privileges provided to them by the input the object and get an output of the correct detection and recognition.

1.2.5 Operating Environment

- OS : Windows 7+, Linux (All Flavors), Mac
- Browser : Chrome, Microsoft Edge, Firefox, any versions supporting HTML5
- Backend : Anaconda Navigator IDE with packages in requirements of Algorithm
- Language : Python 3.8+

1.2.6 Design and Implementation Constraints

- Algorithm will work if requirements mentioned in 1.2.5 are satisfied
- Webpage will detects and classifies object as long as input objects are provided correctly

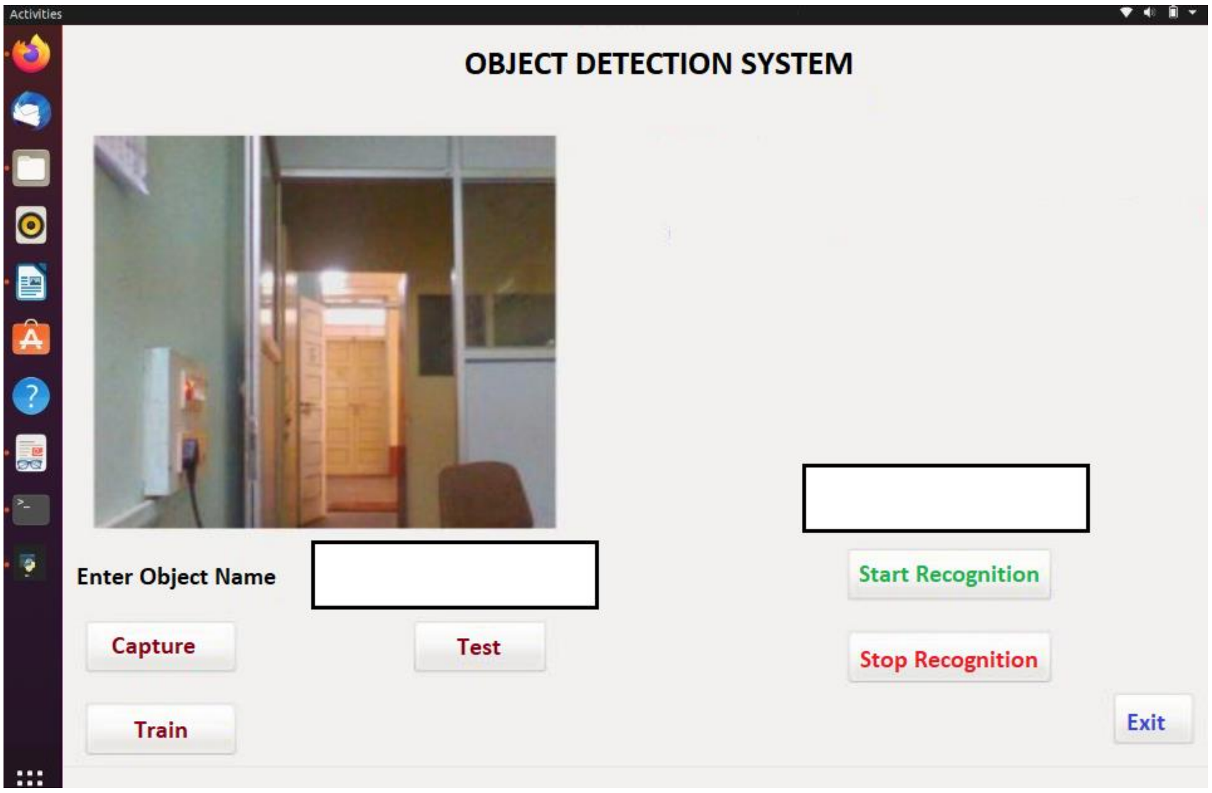
- Algorithm fails to predict correctly in case of features provided which it’s not trained for
- Algorithm is trained on Animals and Students dataset
- If system isn’t supporting HTML protocols, then the webpage won’t work
- Live Server or equivalent tool is expected to run algorithm concurrently

1.2.7 Assumptions and Dependencies

- This detection and prediction is totally based on the trained algorithm
- Machine Learning model must be loaded prior to recognition
- All the features entered are filtered before recognition of object by the algorithm
- Any anomalies in entering values to features are discarded and may affect the result

1.3. External Interface Requirements

1.3.1 User Interface



1.3.2 Hardware Interface

a) Server side

- The web application will be hosted on a web server which is listening on the web standard port, port 80
- Cache memory is to be available
- Network Interface card is to be available
- Wi-Fi Card is optional
- Server will send request to the model to retrieve the results only if hardware and software requirements are met

b) Client side

- Monitor screen - The software shall display information to the user via the monitor screen

- Mouse - The software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus
- Keyboard - The software shall interact with the keystrokes of the keyboard. The keyboard will input data into the model
- The software is built to run on both Windows 32bit or 64bit OS
- RAM or ROM memory able to handle and provide the computational power for the algorithm

1.3.3 Software Interface

a) Server side

- An Apache web server or localhost or VS Code Live Server
- Accept all requests from the client and reply back to it accordingly
- Model will be loaded onto the browser once session instance is initiated
- Response headers supported by the protocol
- Status codes shown successfully and accordingly to the request & responses

b) Client side

- An OS which is capable of running a web browser which supports JavaScript and HTML5
- Model is to be trained and tested ready
- Dataset available on local system
- Source code available to clean the incoming data input from server side
- Necessary request methods supported by the language in which algorithm is coded in and trained and tested

1.3.4 Communication Interfaces

- The HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server
- NIC Card or Wi-Fi Card available

- Network adapters and drivers installed into the system which are able to detect and communicate to internet if required

1.3.5 System Interface

The webpage is going to use local server through VS Code Live Server or Apache - Tomcat Web/Application Server. The user shall provide inputs through the HTML forms.

1.4 System Features

1.4.1 Predict image class

Use Case	**Predict class of image**
Description	Recognizes the object based on the input feature object
Actors	Below mentioned sectors • Agriculture (Farmers) • Organizations (Employee) • Security Force Etc.
Pre-Condition	The user must input all the features and click detect Button
Post-Condition	Trained model is fed the data collected from the fields on the form
Steps	1. The user provided data is fed to the Trained Model 2. Model will compute the detected object 3. Object name is output to the user

1.4.2 Train/Test Model

Use Case	**Train Data/ Test Data**
Description	Builds Machine Learning model based on the input features values stored in dataset
Actors	Developer
Pre-Condition	The user must feed the dataset into program
Post-Condition	Appropriate algorithm, practices are applied
Steps	1. The developer shall feed the dataset into the program 2. Developer must clean the data with appropriate methods and make data ready to train/test on 3. Developer shall apply suitable Machine Learning Algorithm after splitting data into train and test subsets 4. Trained model is ready to give predictions

1.5 Other Nonfunctional Requirements

1.5.1 Performance Requirements

- Static Requirements:
    - Each instance of live server opened is independent of each other, one's input values do not affect other instance's output
    - Each of the server will use a port for requests and responses, make sure to not open lot of instances and exhaust automatically assigned ports

    - Can manually assign a port or free occupied port
    - GUI/ RAM should be sufficient in case of training huge amount of data
- Dynamic Requirements:
    - Model should predict the result within 5seconds
    - Model accuracy should be approximately above 70%

1.5.2 Safety/Security Requirements

- No data collected is shared anywhere
- No authentication is required to use the system
- Only Developer can retrain model on new dataset collected
- User can only get prediction based on input features to already trained model

1.5.3 Software Quality Attributes

- Availability: The webpage is available to anyone once hosted or if the user has the source code with them and live server is running
- Adaptability: The algorithm is adaptable to any type of anomalies fed as input, it will filter them out with appropriate values
- Scalability: The algorithm can be scaled to any extend, the system requirements for computational power must be met prior to it
- Correctness: The correctness of the output will totally depend upon the data the model was trained upon
- Portability: The software is portable to any system meeting the requirements
- Maintainability: There is zero to none maintenance required once the model is trained and webpage is hosted. Only when latest dataset is to be updated then we need to train and test the model again.

2) System Design

2.1 Architecture Diagram

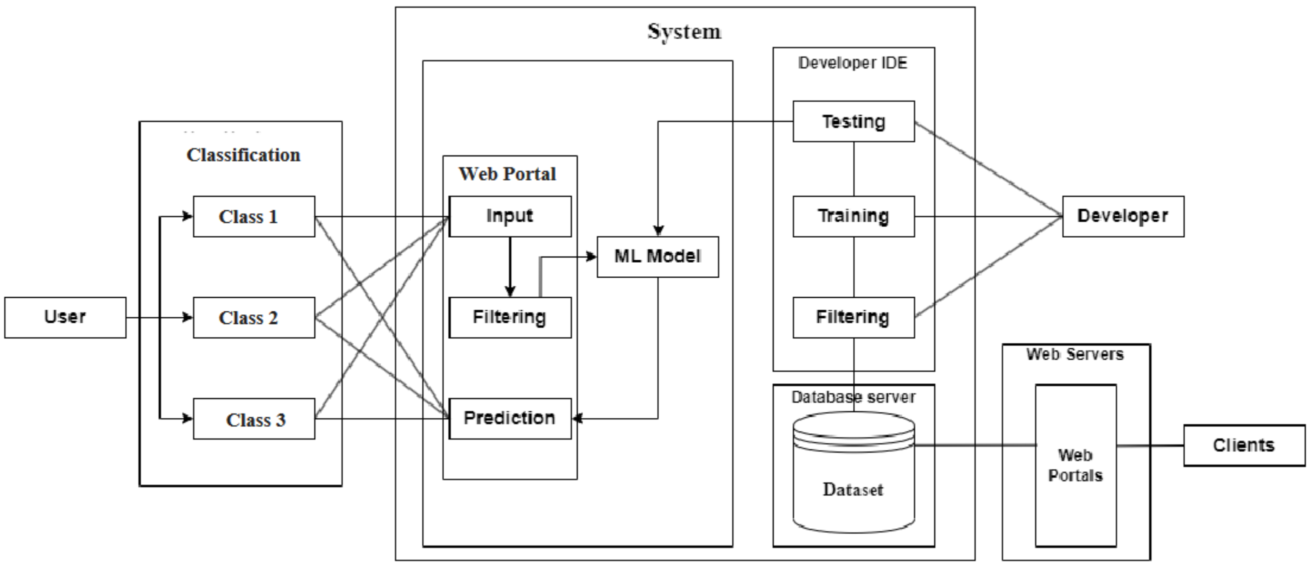


Fig 2.1 Architecture Diagram

The system will be up and running once hosted available to various focus group including the clients, the users and developers. The data to the system is collected from client's input. The developer can then filter the data and apply business logic to obtain the set of features they need to improve overall accuracy of final outcome. The user can provide features as input and obtain the output through the ML Model.

2.2 System Models

2.2.1 PredictClass Module

- The end-user who is interested to use the system will connect to the software through a web portal
- They will be able to provide the features as input through HTML form on portal to the ML Model
- The features are checked and cleaned before being sent to the ML Model
- The model will predict the class of image based on the features and return the results

2.2.2 TrainTest Module

- The developer will create the ML Model through their IDE.
- Once the cleaned data is obtained the developer applies various machine learning techniques to enhance the features
- Once the features are ready, they are split into training and testing data subsets
- The Model is trained on the training subset and later tested on the test subset
- The process is repeated until satisfactory results are obtained i.e., accuracy of model has improved

2.2.3 CollectData Module

- The data is collected through various web portals from potential clients.
- The collected data is in all forms and needs to be organized before being stored into the database
- The data is validated and then organized into labelled data
- This data is cleaned to replace any anomalies and remove outliers that are visible on first glance

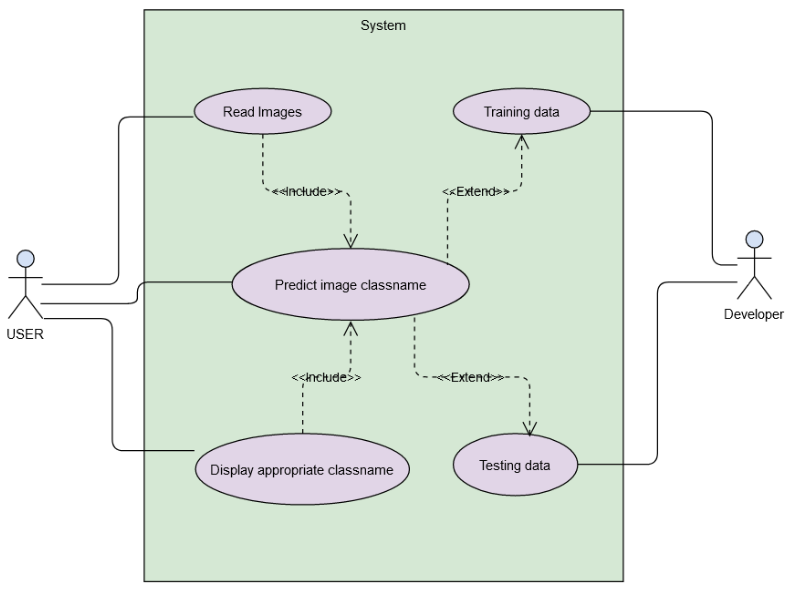
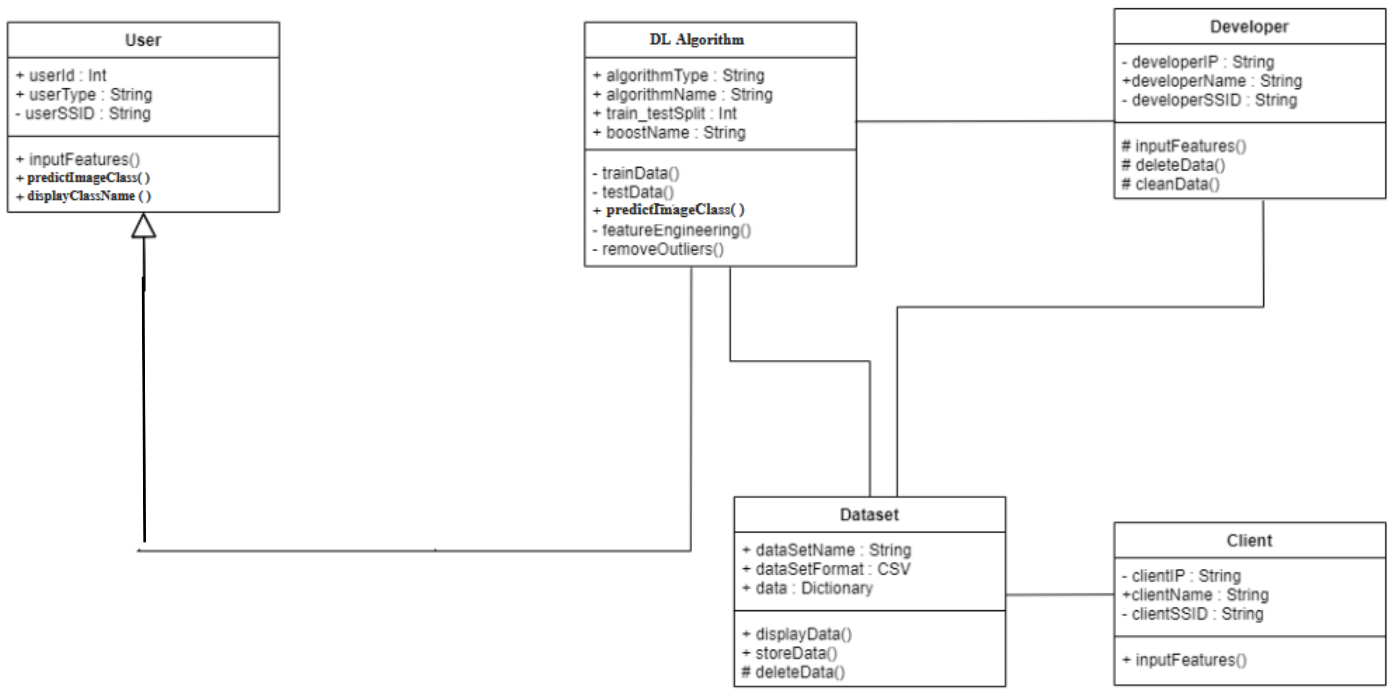


Fig 2.3 Use Case Diagram

3. Detailed Design

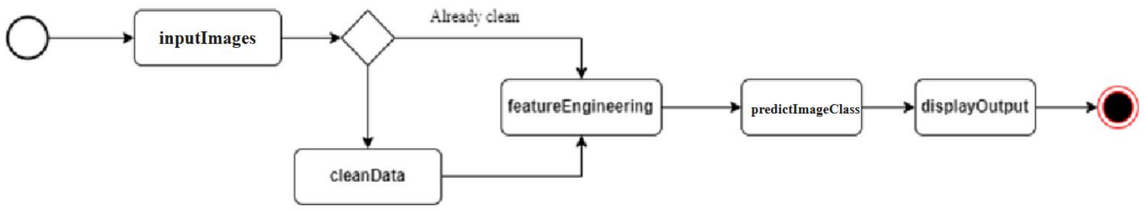
3.1 Class diagram



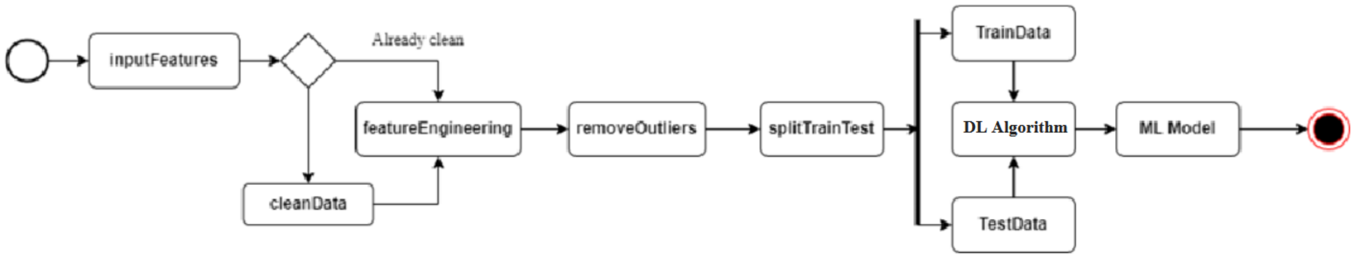
- The Developer class is capable to deleting the data, inputting data, training and testing the ML Algorithm to obtain the ML Model
- The Client Class is only capable of inputting the data
- The Dataset Class will store data, display and only authorized classes are able to use deleteData() method

3.2 Activity Diagram

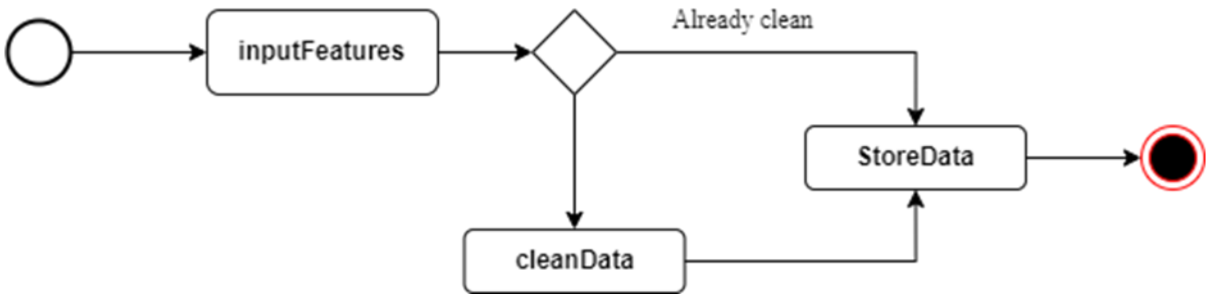
3.2.1 Predict class



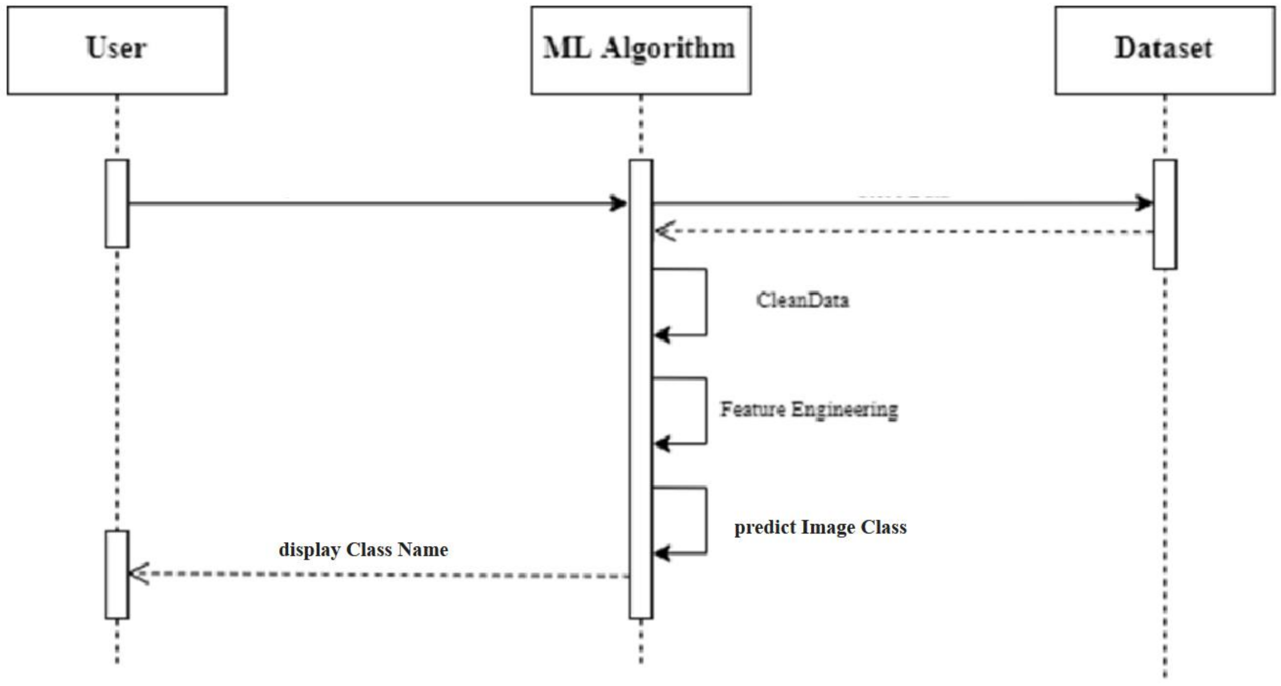
3.2.1 Train Test Data



3.2.3 Collect data

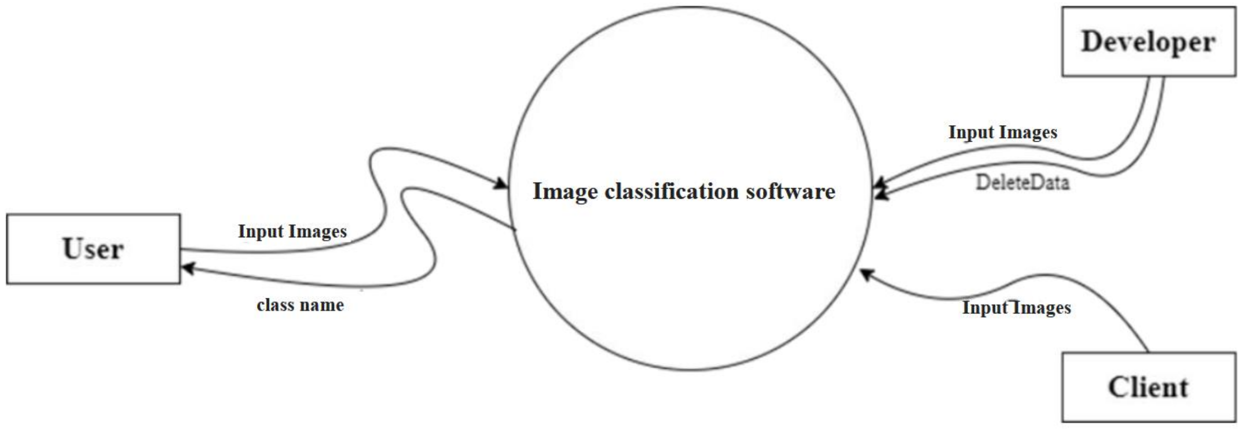


3.3 Sequence Diagram

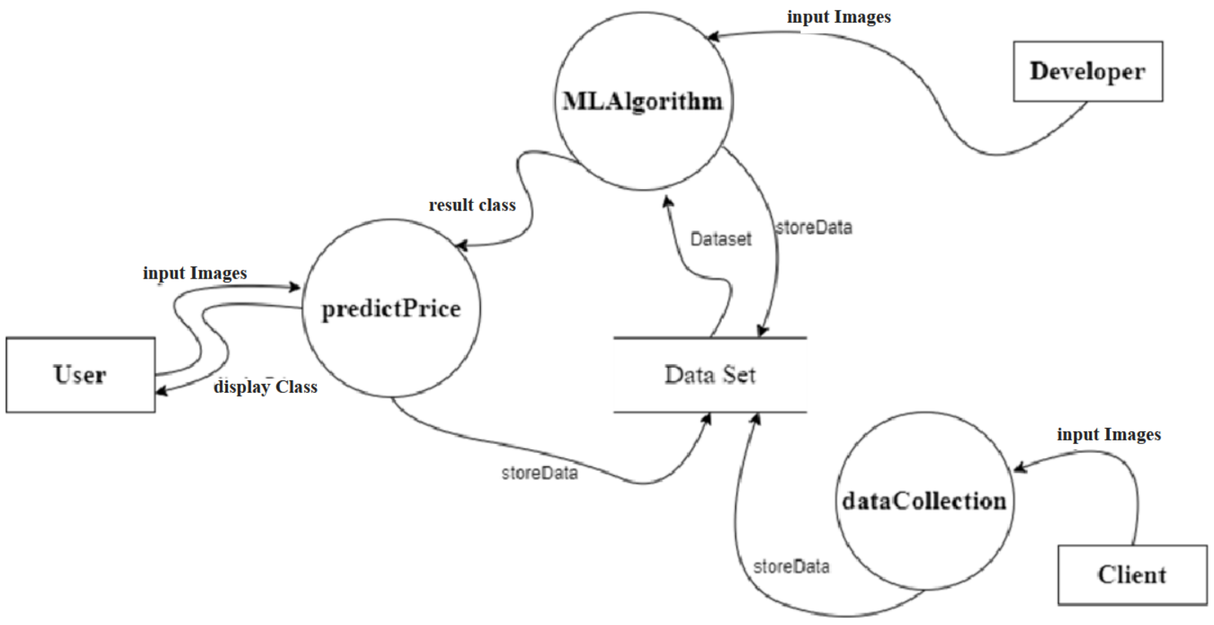


3.4 Data flow diagram

3.4.1 Level 0 DFD



3.4.2 Level 1 DFD



4. Software Testing

4.1 Unit Testing

Sr No.	Testcase Name	Test Description	Expected Input/Output	Actual Input/Output	Testcase Result
1	User Inputs for predicting class of image	Predicting the class of image	Display Correct class name	If any input other than the specified form If no input is provided to a single/few textbox	Fail Pass

2	Dataset path	Must provide a valid/existing dataset path to the algorithm	Valid absolute/relative path to the dataset	Valid absolute/relative path to the dataset Invalid absolute/relative path to the dataset	Pass Fail
3	TrainTest Split value	Must enter the train test split value in ratio of 3:1 or 70% to 30%	Training set percentage: Integer Testing set percentage: Integer	Training set percentage: Integer Testing set percentage: Integer Any value other than Integer or ratio	Pass Fail

4.2 Integration Testing

Sr No.	Testcase Name	Test Description	Expected Input/Output	Actual Input/Output	Testcase Result
1	TrainTest Data	The following constraints are to be provided when training and testing dataset • Import necessary packages • Algorithm used for analysis • Features must be labelled	• Imported packages needed • Valid Algorithm • Features are labelled	• No valid packages are imported • No valid algorithm used • Features are not labelled • Valid packages are imported • Valid algorithm used • Features are labelled	Fail Fail Fail Pass Pass Pass

4.3 System Testing

Sr No.	Testcase Name	Test Description	Expected Input/Output	Actual Input/Output	Testcase Result
1	Image class Prediction	Successfully give the class of the image	Class is predicted successfully	Price returned is not of actual class Price returned is of correct class Undefined	Fail Pass Fail
2	Store prepared dataset	The dataset on which feature engineering is applied after cleaning it and removing all outliers is to be stored successfully	Dataset stored successfully	Dataset stored successfully Dataset is not stored	Pass Fail
3	MLModel	Successfully train and test the ML Model	Successfully train and test the ML Model based on valid algorithm and obtain satisfactory performance for our Task	Successfully trained and tested the ML Model based on valid algorithm and obtained satisfactory performance for our Task	Pass

				Unsuccessful in training and testing the ML Model obtaining satisfactory performance for our Task	Fail

5.References

https://ece.anits.edu.in/

https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a

https://www.tensorflow.org/lite/examples/object_detection/overview

https://towardsdatascience.com/object-detection-with-tensorflow-model-and-opencv-d839f3e42849

https://github.com/tensorflow/models/tree/master/research/object_detection