A PROJECT REPORT ON

BE PROJECT TITLE

SUBMITTED TOWARDS THE
PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Shubham Zope	Exam No: B150134351
Abhijit Watpade	Exam No: B150134350
Sanket Sonar	Exam No: B150134331
Sameer Rathod	Exam No: B150134319

Under The Guidance of

Prof. Guide Name

Department of Computer Engineering
K. K. Wagh Institute of Engineering Education & Research
Hirabai Haridas Vidyanagari, Amrutdham, Panchavati,
Nashik-422003
Savitribai Phule Pune University

A. Y. 2020-21 Sem I

K. K. Wagh Institute of Engineering Education and Research
Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Titled

BE PROJECT TITLE

Submitted by

Shubham Zope	Exam No: B150134351
Abhijit Watpade	Exam No: B150134350
Sanket Sonar	Exam No: B150134331
Sameer Rathod	Exam No: B150134319

is a bonafide work carried out by Students under the supervision of Prof. Guide Name and it is submitted towards the partial fulfilment of the requirement of Bachelor of Engineering (Computer Engineering) Project during academic year 2020-21.

Prof. Guide Name	Prof. Dr. S. S. Sane
Internal Guide	Head
Department of Computer Engineering	Department of Computer Engineering

ABSTRACT

Sign language is one of the oldest and most natural form of language for communication, hence we have come up with a real time method using neural networks for finger spelling based American sign language. Automatic human gesture recognition from camera images is an interesting topic for developing vision. We propose a convolution neural network (CNN) method to recognize hand gestures of human actions from a image captured by camera. The purpose is to recognize hand gestures of human task activities from a camera image. The skin model, position of hand and orientation are applied to obtain the training and testing data for the CNN. The hand is first passed through a filter and after the filter is applied where the hand is passed through a classifier which predicts the class of the hand gestures. The hand position aims at translating and rotating the hand image to a neutral pose. Then the calibrated images are used to train the CNN.

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on
'SIGN LANGUAGE TO SPEECH CONVERSION USING CONVOLUTIONAL NEURAL NETWORK'

We would like to take this opportunity to thank our internal guide **Prof. N .M. Shahane** for giving us all the help and guidance we needed. We are really grateful to them for his kind support. His valuable suggestions were very helpful.

We are also grateful to **Prof. S.S Sane**, Head of Computer Engineering Department,K.K.Wagh Institute Of Engineering Education And Research, Nashik for his indispensable support, suggestions.

In the end our special thanks to **Ms. I. Priyadarshini** for providing various resources such as arranging online meetings, continous interaction with us.

Shubham Zope
Abhijit Watpade
Sanket Sonar
Sameer Rathod
(B.E. Computer Engg.)

INDEX

1 Introduction **1**

1.1 Project Idea 2

1.2 Motivation of the Project 2

1.3 Literature Survey 3

2 Problem Definition and scope **6**

2.1 Problem Statement 7

2.1.1 Goals and objectives 7

2.1.2 Statement of scope 7

2.2 Major Constraints 7

2.3 Methodologies of Problem solving and efficiency issues 8

2.4 Hardware Resources Required 10

2.5 Software Resources Required 10

3 Project Plan **11**

3.0.1 Reconciled Estimates 12

3.0.2 Project Resources 12

3.1 Risk Management 12

3.1.1 Risk Identification 12

3.1.2 Risk Analysis 13

3.1.3 Overview of Risk Mitigation, Monitoring, Management . . 13

3.2 Project Schedule 15

3.2.1 Project task set 15

3.2.2 Timeline Chart 15

3.3 Team Organization 15
3.3.1 Team structure 16
3.3.2 Management reporting and communication 16

4 Software requirement specification 17
4.1 Introduction 18
4.1.1 Purpose and Scope of Document 18
4.2 Usage Scenario 18
4.2.1 Use-cases 18
4.2.2 Use Case View 18
4.3 Data Model and Description 18
4.3.1 Data Description 18
4.3.2 Data objects and Relationships 19
4.4 Functional Model and Description 19
4.4.1 Data Flow Diagram 19
4.4.2 Description of functions 19
4.4.3 Activity Diagram: 20
4.4.4 Non Functional Requirements: 20
4.4.5 State Diagram: 21
4.4.6 Design Constraints 21
4.4.7 Software Interface Description 21

5 Detailed Design Document 23
5.1 Introduction 24
5.2 Architectural Design 24
5.3 Data design 24
5.3.1 Internal software data structure 24
5.3.2 Global data structure 25
5.3.3 Temporary data structure 25
5.3.4 Database description 25

6 Dataset and Experimental setup 26

7 Summary and Conclusion **27**

Annexure A Mathematical Model **31**

Annexure B Plagiarism Report **32**

Annexure C Paper Published (if any) **33**

Annexure D Sponsorship detail (if any) **34**

List of Figures

4.1 Use case diagram 19
4.2 Activity diagram 20
4.3 State transition diagram 22

5.1 Architecture diagram 24

List of Tables

2.1 Hardware Requirements 10

3.1 Risk Table 13
3.2 Risk Probability definitions 13
3.3 Risk Impact definitions [1] 14

4.1 Use Cases 18

CHAPTER 1

INTRODUCTION

1.1 PROJECT IDEA

- To use a web scrapper to recognize sign from images and create local database of images using Convolutional Neural Networks (CNN) to augment the limited number of local images available.

1.2 MOTIVATION OF THE PROJECT

- While the world has progressed in online interaction through various tools, but the D&M (Dumb & Deaf) people still face a language barrier.So, they depend on vision based communication for interaction.
- If there is a common interface that converts the sign language to text the hand gestures can be easily understood by the other people.
- So research has been made for a vision based interface system where D&M people can enjoy the communication without really knowing each other's language.
- The aim is to develop a user friendly sign to speech conversion software where the computer understands the human sign language.
- There are different sign languages all over the world, namely American Sign Language (ASL), French Sign Language, British Sign Language (BSL), Indian Sign language, Japanese Sign Language and work has been done on other languages all around the world.

1.3 LITERATURE SURVEY

In the recent years there has been tremendous research done on the hand gesture recognition.
With the help of literature survey done we realized the basic steps in hand gesture recognition are :-

- Data acquisition

  The different approaches to acquire data about the hand gesture can be done in the following ways:

  - Use of sensory devices It uses electronically devices to provide exact hand configuration, and position. Different glove based approaches can be used to extract information .But it is expensive and not user friendly.
  - Vision based approach In vision based methods computer camera is the input device for observing the information of hands or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing 7 artificial vision systems that are implemented in software and/or hardware. The main challenge of vision-based hand detection is to cope with the large variability of human hand's appearance due to a huge number of hand movements, to different skin-colour possibilities as well as to the variations in view points, scales, and speed of the camera capturing the scene.

- Data pre processing and Feature extraction

  - In [1] the approach for hand detection combines threshold-based color detection with background subtraction.We can use Adaboost face detector to differentiate between faces and hands as both involve similar skin-color
  - We can also extract necessary image which is to be trained by applying

a filter called Gaussian blur. The filter can be easily applied using open computer vision also known as OpenCV and is described in [3].

- For extracting necessary image which is to be trained we can use instrumented gloves as mentioned in [4]. This helps reduce computation time for preprocessing and can give us more concise and accurate data compared to applying filters on data received from video extraction.
- We tried doing the hand segmentation of an image using color segmentation techniques but as mentioned in the research paper skin color and tone is highly dependent on the lighting conditions due to which output we got for the segmentation we tried to do were no so great. Moreover we have a huge number of symbols to be trained for our project many of which look similar to each other like the gesture for symbol 'V' and digit '2', hence we decided that in order to produce better accuracies for our large number of symbols, rather than segmenting the hand out of a random background we keep background of hand a stable single color so that we don't need to segment it on the basis of skin color . This would help us to get better results.

- Gesture classification

  - In [1] Hidden Markov Models (HMM) is used for the classification of the gestures .This model deals with dynamic aspects of gestures.Gestures are extracted from a sequence of video images by tracking the skin-colour blobs corresponding to the hand into a body– face space centered on the face of the user. The goal is to recognize two classes of gestures: deictic and symbolic.The image is filtered using a fast look–up indexing table. After filtering, skin colour pixels are gathered into blobs. Blobs are statistical objects based on the location (x,y) and the colourimetry (Y,U,V) of the skin colour pixels in order to determine homogeneous areas

    [2] Naïve Bayes Classifier is used which is an effective and fast method for static hand gesture recognition. It is based on classifying the differ-

ent gestures according to geometric based invariants which are obtained from image data after segmentation.Thus,unlike many other recognition methods, this method is not dependent on skin colour. The gestures are extracted from each frame of the video,with a static background. The first step is to segment and label the objects of interest and to extract geometric invariants from them. Next step is the classification of gestures by using a K nearest neighbor algorithm aided with distance weighting algorithm (KNNDW) to provide suitable data for a locally weighted Naïve Bayes classifier.

– According to paper on "Human Hand Gesture Recognition Using a Convolution Neural Network" by Hsien-I Lin , Ming-Hsiang Hsu, and Wei-Kai Chen graduates of Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan, they construct a skin model to extract the hand out of an image and then 9 apply binary threshold to the whole image. After obtaining the threshold image they calibrate it about the principal axis in order to center the image about it. They input this image to a convolutional neural network model in order to train and predict the outputs. They have trained their model over 7 hand gestures and using their model they produce an accuracy of around 95% for those 7 gestures.

CHAPTER 2

PROBLEM DEFINITION AND SCOPE

2.1 PROBLEM STATEMENT

To create a computer application and train a model which when shown a real time video of hand gestures of American Sign Language shows the output for that particular sign in text format then gets converted into audio format.

2.1.1 Goals and objectives

Goal and Objectives:

- To create a completely functional product for the people who are not able to hear, so that, they can get connected to the world easily.
- To use and understand technologies like OpenCV, Matplotlib, Keras, Deep Learning, Python, Heroku Host, etc.
- To create a web based project for detecting and understanding American Sign Language using Machine Learning Concepts.
- To create own Database for training and testing CNN Model
- To reduce the gap of technology between the deaf and dump people with this web based application.

2.1.2 Statement of scope

- Input image size is 128 * 128 Pixels.
- Input image format JPG.
- The sign detection does not work for word level sign vocabulary and Facial Expressions.

2.2 MAJOR CONSTRAINTS

- The end user needs to capture the sign of the letter which he/she wants to translate into audio

- The sign detection does not work for word level sign vocabulary and Facial Expressions.

2.3 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

- The system is a vision based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction.

- **Data Set Generation**

  For the project we tried to find already made datasets but we couldn't find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence we decided to create our own data set. Steps we followed to create our data set are as follows.

  We have decied to use Open computer vision(OpenCV) library in order to produce our dataset.Firstly we are going to capture around 800 images of each of the symbol in ASL for training purposes and around 200 images per symbol for testing purpose. First we capture each frame shown by the webcam of our machine. In the each frame we define a region of interest (ROI) which is denoted by a blue bounded square.

  From this whole image we extract our ROI which is RGB and convert it into gray scale Image.

  Finally we apply our gaussian blur filter to our image which helps us extracting various features of our image. The image after applying gaussian blur.

  GESTURE CLASSIFICATION :-

  Algorithm Layer :-

  1. Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.

  2. This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken

into consideration for forming the word.

3. Space between the words are considered using the blank symbol.

Activation Function :-

We have decided to use ReLU(Rectified Linear Unit) as our activation function in each layer. It is a simple function stated as **f(x) = max(0,x)** for each input pixel.Using these activation function we would be able to reduce the required computation power , Gaussian Descent losing problems, etc.

Pooling Layer :-

We apply Max pooling to the input image with a pool size of (2, 2) with relu activation function.This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

Optimizer :-

We are going to use Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm(ADA GRAD) and root mean square propagation(RMSProp)

Finger spelling sentence formation Implementation :-

1. Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold we print the letter and add it to the current string(In our code we kept the value as 50 and difference threshold as 20).

2. Otherwise we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.

3. Whenever the count of a blank(plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.

4. In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

Autocorrect Feature :-

A python library Hunspell suggest is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence.This helps in reducing mistakes committed in spellings and assists in predicting complex words.

2.4 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	Core i5 2.6GHz	For Training Model
2	RAM	4 GB	To load large dataset
3	GPU	GTX 1050 2GB	To train model with GPU parallelism

Table 2.1: Hardware Requirements

2.5 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System: Windows 8 and Above
2. IDE: Pychamp
3. Programming Language: Python 3.6
4. Python libraries: OpenCV, Numpy, Matplotlib, Keras, PIL

CHAPTER 3

PROJECT PLAN

3.0.1 Reconciled Estimates

3.0.1.1 Time Estimates

$$C = D * Cp * hrs$$

Where, C = Cost of project

D = Duration in Hours

Cp = Cost incurred per person-hour

hrs = hours

Total of 4.5 person-months are required to complete the project successfully.

Duration of Project D = 6 Months

The approximate duration of the project is 4 months

3.0.2 Project Resources

Project resources [People, Hardware, Software, Tools and other resources] based on Memory Sharing, IPC, and Concurrency derived using appendices to be referred.

3.1 RISK MANAGEMENT

This section discusses Project risks and the approach to managing them.

3.1.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [1]. Please refer table 3.1 for all the risks. You can refereed following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project?

2. Are end-users enthusiastically committed to the project and the system/product to be built?

3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end-users have realistic expectations?
6. Does the software engineering team have the right mix of skills?
7. Are project requirements stable?
8. Is the number of people on the project team adequate to do the job?
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

3.1.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Description 1	Low	Low	High	High
2	Description 2	Low	Low	High	High

Table 3.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 3.2: Risk Probability definitions

3.1.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 3.3: Risk Impact definitions [1]

Risk ID	1
Risk Description	Description 1
Category	Development Environment.
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Strategy
Risk Status	Occurred

Risk ID	2
Risk Description	Description 2
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Better testing will resolve this issue.
Risk Status	Identified

Risk ID	3
Risk Description	Description 3
Category	Technology
Source	This was identified during early development and testing.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Example Running Service Registry behind proxy balancer
Risk Status	Identified

3.2 PROJECT SCHEDULE

3.2.1 Project task set

Major Tasks in the Project stages are:

- Task 1:
- Task 2:
- Task 3:
- Task 4:
- Task 5:

3.2.2 Timeline Chart

A project timeline chart is presented. This may include a time line for the entire project. Above points should be covered in Project Planner as Annex C and you can mention here Please refer Annex C for the planner

3.3 TEAM ORGANIZATION

- Team of 4 members
- 1 Project Guide

- 1 Project Coordinator

3.3.1 Team structure

Team of 4 members

3.3.2 Management reporting and communication

- Online meeting with guide about project work
- Project Idea Presentation
- Progress Presentation (1st Semester)

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

4.1.1 Purpose and Scope of Document

The purpose of the project is to create computer application and train a model that captures images of hand gestures of American Sign Language and shows the output for particular sign in audio format.

4.2 USAGE SCENARIO

4.2.1 Use-cases

All use-cases for the software are presented. Description of all main Use cases using use case template is to be provided.

Sr No.	Use Case	Description	Actors	Assumptions
1	Use Case 1	Description	Actors	Assumption
2	height			

Table 4.1: Use Cases

4.2.2 Use Case View

Use Case Diagram. Example is given below

4.3 DATA MODEL AND DESCRIPTION

4.3.1 Data Description

Data objects that will be managed/manipulated by the software are described in this section. The database entities or files or data structures required to be described. For data objects details can be given as below

Figure 4.1: Use case diagram

4.3.2 Data objects and Relationships

Data objects and their major attributes and relationships among data objects are described using an ERD- like form.

4.4 FUNCTIONAL MODEL AND DESCRIPTION

A description of each major software function, along with data flow (structured analysis) or class hierarchy (Analysis Class diagram with class description for object oriented system) is presented.

4.4.1 Data Flow Diagram

4.4.1.1 Level 0 Data Flow Diagram

4.4.1.2 Level 1 Data Flow Diagram

4.4.2 Description of functions

A description of each software function is presented. A processing narrative for function n is presented.(Steps)/ Activity Diagrams. For Example Refer 4.2

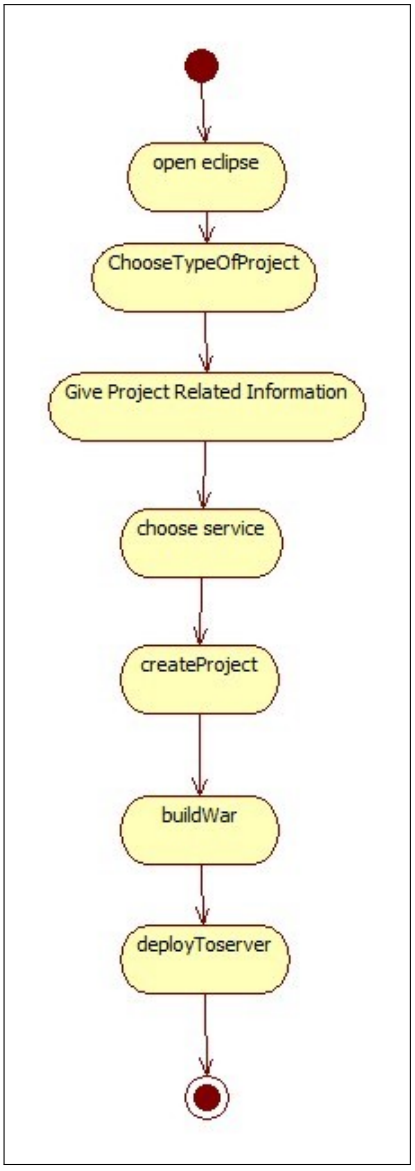

Figure 4.2: Activity diagram

4.4.3 Activity Diagram:

- The Activity diagram represents the steps taken.

4.4.4 Non Functional Requirements:

- Interface Requirements
- Performance Requirements
- Software quality attributes such as availability [related to Reliability], modi-

fiability [includes portability, reusability, scalability] , performance, security, testability and usability[includes self adaptability and user adaptability]

4.4.5 State Diagram:

State Transition Diagram

Fig.4.3 example shows the state transition diagram of Cloud SDK. The states are represented in ovals and state of system gets changed when certain events occur. The transitions from one state to the other are represented by arrows. The Figure shows important states and events that occur while creating new project.

4.4.6 Design Constraints

Any design constraints that will impact the subsystem are noted.

4.4.7 Software Interface Description

The software interface(s)to the outside world is(are) described. The requirements for interfaces to other devices/systems/networks/human are stated.

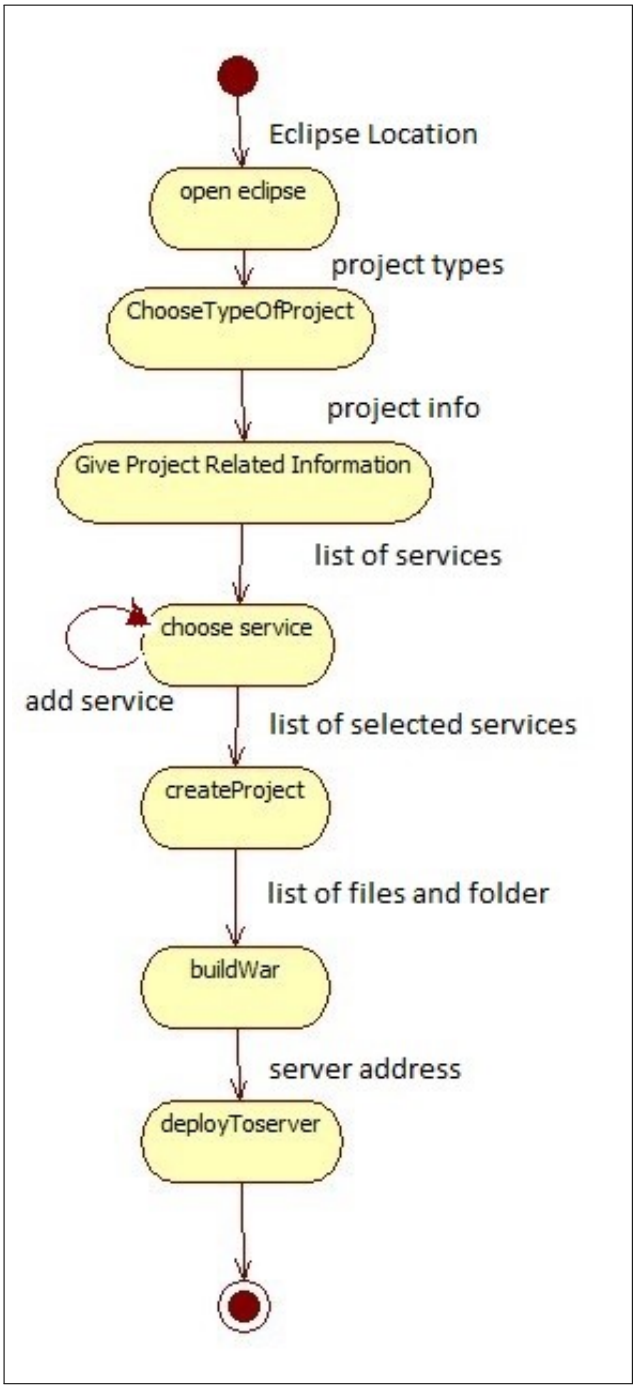

Figure 4.3: State transition diagram

CHAPTER 5

DETAILED DESIGN DOCUMENT

5.1 INTRODUCTION

This document specifies the design that is used to solve the problem of Product.

5.2 ARCHITECTURAL DESIGN

A description of the program architecture is presented. Subsystem design or Block diagram,Package Diagram,Deployment diagram with description is to be presented.

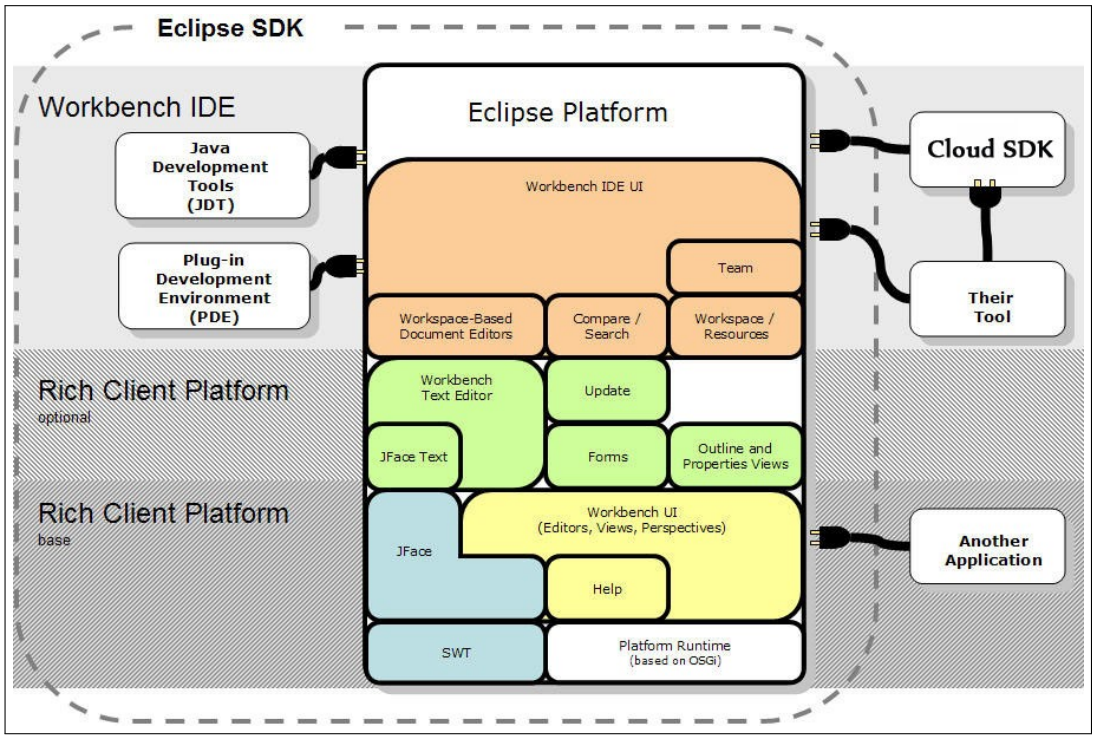

Figure 5.1: Architecture diagram

5.3 DATA DESIGN

A description of all data structures including internal, global, and temporary data structures, database design (tables), file formats.

5.3.1 Internal software data structure

Data structures that are passed among components the software are described.

5.3.2 Global data structure

Data structured that are available to major portions of the architecture are described.

5.3.3 Temporary data structure

Files created for interim use are described.

5.3.4 Database description

Database(s) / Files created/used as part of the application is(are) described.

CHAPTER 6

DATASET AND EXPERIMENTAL SETUP

CHAPTER 7

SUMMARY AND CONCLUSION

In this report, a functional real time vision based American sign language recognition for deaf and dumb people have been developed for ASL alphabets.We can achieve final accuracy of 98.0 percent on our data set. We would be able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

REFERENCES

[1] R. S. Pressman, *Software Engineering (3rd Ed.): A Practitioner's Approach*. New York, NY, USA: McGraw-Hill, Inc., 1992.

ANNEXURE A

MATHEMATICAL MODEL

ANNEXURE B

PLAGIARISM REPORT

ANNEXURE C

PAPER PUBLISHED (IF ANY)

ANNEXURE D

SPONSORSHIP DETAIL (IF ANY)