**PROJECT REPORT**

**On**

**Signature Authentication in Air**

Submitted to Rajasthan Technical University

in partial fulfillment of the requirement for the award of the degree of

**B. Tech.**

**in**

**Computer Science and Engineering**

**Submitted By**

**Abhishek Sharma (PIET15CE006)**

**Ankit Kumawat (PIET15CE017)**

**Lokesh Choudhary (PIET15CE055)**

**Under the Guidance of**

**Mr. Deepak Moud**

at



**POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY, JAIPUR**

**Rajasthan Technical University, KOTA**

**APRIL, 2019**

**CERTIFICATE**

This is to be certified that the project entitled “**Signature Authentication in Air** ” has been submitted for the Bachelor of Computer Science and Engineering, Poornima Institute of Engineering & Technology, Jaipur during the academic year 2018-2019 is a bonafide piece of project work carried out by “ **Abhishek Sharma, Ankit Kumawat & Lokesh Choudhary**” towards the partial fulfillment for the award of the Degree (B.Tech.) under the guidance of “**Mr. Deepak Moud**” and supervision and no part of thereof has been submitted by them for any degree or diploma.

Project Guide Project Coordinator Mr. Deepak Moud

Mr. Deepak Moud Prof. (Dr.) Praveen Gupta (H.O.D,CSE)

(Designation) (Professor)

**CANDIDATE’S DECLARATION**

We, **Abhishek Sharma (PIET15CE006), Ankit Kumawat(PIET15CE017) & Lokesh Choudhary(PIET15CE055)** B.Tech (Semester- VIII) of “**Poornima Institute of Engineering & Technology, Jaipur”** hereby declare that the Project Report entitled **“Signature Authentication in Air”** is an original work and data provided in the study is authentic to the best of our knowledge.This report has not been submitted to any other Institute for the award of any other degree.

|  |  |  |
| --- | --- | --- |
| **Abhishek sharma** | **ankit kumawat** | **lokesh choudhary** |
| **(PIET15CE006)** | **(PIET15CE017)** | **(PIET15CE055)** |

|  |  |
| --- | --- |
| **Place: Jaipur** |  |
| **Date: 06/03/2019** |  |

**ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to ***Dr. Dinesh Goyal,*** Principal, PIET for providing us an opportunity to undergo this Major Project as the part of the curriculum.

We are thankful to ***Mr. Deepak Moud, HOD, CS*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We are thankful to ***Mr. Deepak Moud*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to ***Prof. (Dr.) Praveen Gupta*** who provided his valuable suggestions and precious time in accomplishing our Project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

|  |  |  |
| --- | --- | --- |
| **Abhishek Sharma** | **Ankit Kumawat** | **Lokesh Choudhary** |
| **( PIET15CE006 )** | **(PIET15CE017 )** | **(PIET15CE055)** |

**TABLE OF CONTENTS**

Contents

[LIST OF FIGURES](#_Toc614613) iii

[ABSTRACT iv](#_Toc614614)

[KEYWORDS: iv](#_Toc614615)

[CHAPTER 1 1](#_Toc614616)

[INTRODUCTION TO PROJECT 1](#_Toc614617)

[Project Aim and Objective 1](#_Toc614618)

[Problem Statement 1](#_Toc614619)

[Literature Review 2](#_Toc614620)

[Software Requirements 2](#_Toc614621)

[Hardware Requirements 12](#_Toc614622)

[CHAPTER 2 13](#_Toc614623)

[PRODUCT BACKLOG 13](#_Toc614624)

[Product Backlog 13](#_Toc614625)

[Sprint Backlog-1 16](#_Toc614626)

[Sprint Backlog-2 21](#_Toc614627)

[Sprint Backlog-3 25](#_Toc614628)

[Sprint Backlog-4 30](#_Toc614629)

[CHAPTER 3 33](#_Toc614630)

[TECHNOLOGY APPLIED AND PROJECT MANAGEMENT 33](#_Toc614631)

[Project Management : 33](#_Toc614632)

[Software Project Management 33](#_Toc614633)

[Software Project 34](#_Toc614634)

[Need of software project management 34](#_Toc614635)

[Software Project Manager 35](#_Toc614636)

[Software Management Activities 35](#_Toc614637)

[Project Planning 36](#_Toc614638)

[Scope Management 36](#_Toc614639)

[Project Estimation 36](#_Toc614640)

[Project Estimation Techniques 37](#_Toc614641)

[Decomposition Technique 37](#_Toc614642)

[Empirical Estimation Technique 38](#_Toc614643)

[Project Scheduling 38](#_Toc614644)

[**Resource Management** 39](#_Toc614645)

[Project Risk Management 39](#_Toc614646)

[Risk Management Process 39](#_Toc614647)

[**Project Execution & Monitoring** 40](#_Toc614648)

[Project Communication Management 40](#_Toc614649)

[Configuration Management 41](#_Toc614650)

[Project Management Tools: 41](#_Toc614651)

[01. Trello 41](#_Toc614652)

[02. Basecamp 41](#_Toc614653)

[03. Teamwork Projects 42](#_Toc614654)

[04. Resource Guru 42](#_Toc614655)

[05. ActiveCollab 42](#_Toc614656)

[06. Zoho Projects 42](#_Toc614657)

[07. Jira 42](#_Toc614658)

[08. Asana 42](#_Toc614659)

[09. Podio 43](#_Toc614660)

[10. Freedcamp 43](#_Toc614661)

[11. Wrike 43](#_Toc614662)

[PO and Their Relevance to project 43](#_Toc614663)

[CHAPTER 4 48](#_Toc614664)

[PROJECT IMPLEMENTATION 48](#_Toc614665)

[Sprint Backlog-1 48](#_Toc614666)

[Sprint Backlog-2 52](#_Toc614667)

[Sprint Backlog-3 55](#_Toc614668)

[Sprint Backlog-4 59](#_Toc614669)

[CHAPTER 5 60](#_Toc614670)

[CONCLUSION 60](#_Toc614671)

[Results 60](#_Toc614672)

[Conclusion 60](#_Toc614673)

[Future Scope 60](#_Toc614674)

[ANNEXURES 61](#_Toc614675)

[www.github.com 61](#_Toc614676)

[( details of github ) 28](#_Toc614677)

[References 61](#_Toc614678)

[Research Paper 61](#_Toc614679)

[DST Document. 61](#_Toc614680)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **FIGURE** | **PAGE NO.** |
| 1. | Burn Down Chart-1 | 18 |
| 2. | Output Spring Backlog-1 | 19 |
| 3. | Burn Down Chart-2 | 23 |
| 4. | Output Spring Backlog-2 | 24 |
| 5. | Burn Down Chart-3 | 28 |
| 6. | Output Spring Backlog-3 | 29 |
| 7. | Burn Down Chart-4 | 32 |
| 8. | Output of Spring Backlog-1 | 51 |
| 9. | Output of Spring Backlog-2 | 54 |
| 10. | Output of Spring Backlog-3 | 55 |
| 11. | Output of Spring Backlog-4 | 59 |

**ABSTRACT**

KEYWORDS: Machine learning, Deep Learning, Signature Authentication

Signature authentication is the process of using machine-learning methods to validate the authenticity of an individual's signature. Aim of this project is to design an algorithm, which can distinguish between genuine and forged signatures using writer independent features, and to develop a system using this algorithm which can be used to verify signatures in organizations like in banks on cheques. We intend to build a complete end-to-end software system which can be used to take signature of the person without physical interaction using camera, perform signature verification, and display the results. For this purpose, various deep learning techniques were developed and tested on standard datasets for signature authentication, as well as on a dataset collected by ourselves.

**CHAPTER 1**

# INTRODUCTION TO PROJECT

## Project Aim and Objective:

Signature verification is a technique used by banks, intelligence agencies and high-profile institutions to validate the identity of an individual. Signature verification is often used to compare signatures in bank offices and other branch capture. An image of a signature or a direct signature is fed into the signature verification software and compared to the signature image on file. In today’s scenario private or public sector, organizations are using signatures to authenticate a person. This process is offline, there is more chances to authenticate a forged signature as genuine, and this process is time taking too and taking human resources. We are developing a solution for this by using machine-learning methods to validate the authenticity of an individual's signature. Aim of this project is to design an algorithm, which can distinguish between genuine and forged signatures using writer independent features, and to develop a system using this algorithm which can be used to verify signatures in organizations like in banks oncheques. We intend to build a complete end-to-end software system which can be used to take signature of the person without physical interaction using camera, perform signature verification, and display the results.

## Problem Statement:

Today almost in every organizations whether it is private or public, signatures are used to check authenticity of that person. This process is done manually which consumes human resources. There are always chances of taking a forged signature as genuine. This takes time too. So the current process is bad in each perspective because it is consuming more time, consuming more human resources. After all this, the accuracy is also less.

## 

## Literature Review :

Signature verification is a type of software that compares signatures and checks for authenticity. This saves time and energy and helps to prevent human error during the signature process and lowers chances of fraud in the process of authentication. The software generates a confidence score against the signature to be verified. Too low of a confidence score means the signature is most likely a forgery.

Signature verification software has now become lightweight, fast, flexible and more reliable with multiple options for storage, multiple signatures against one ID and a huge database. It can automatically search for a signature within an image or file. Following are some research papers used for the reference:

1. Hafemann, Luiz G., Robert Sabourin, and Luiz S. Oliveira. "Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks" <http://dx.doi.org/10.1016/j.patcog.2017.05.012> ([preprint](https://arxiv.org/abs/1705.05787)).
2. .Hafemann, Luiz G., Robert Sabourin, and Luiz S. Oliveira. "Fixed-sized representation learning from Offline Handwritten Signatures of different sizes" <https://doi.org/10.1007/s10032-018-0301-6> ([preprint](https://arxiv.org/abs/1804.00448)).

## SoftwareRequirements:

We need Python 3 installed on the system on which we are going to use the software. We need following packages installed to run the software:

Python-

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

* web development (server-side)
* software development
* mathematics
* System scripting
* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.
* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.
* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.
* Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Math-

1. Selecting the right algorithm which includes giving considerations to accuracy, training time, model complexity, number of parameters and number of features.
2. Choosing parameter settings and validation strategies.
3. Identifying under fitting and over fitting by understanding the Bias-Variance tradeoff.
4. Estimating the right confidence interval and uncertainty.
5. Probability Theory and Statistics: Machine Learning and Statistics aren’t very different fields. Actually, someone recently defined Machine Learning as ‘doing statistics on a Mac’. Some of the fundamental Statistical and Probability Theory needed for ML are Combinatorics, Probability Rules & Axioms, Bayes Theorem, Random Variables, Variance and Expectation, Conditional and Joint Distributions, Standard Distributions (Bernoulli, Binomial, Multinomial, Uniform and Gaussian), Moment Generating Functions, Maximum Likelihood Estimation (MLE), Prior and Posterior, Maximum a Posteriori Estimation (MAP) and Sampling Methods.
6. Multivariate Calculus: Some of the necessary topics include Differential and Integral Calculus, Partial Derivatives, Vector-Values Functions, Directional Gradient, Hessian, Jacobian, Laplacian and Lagragian Distribution.
7. Algorithms and Complex Optimizations: This is important for understanding the computational efficiency and scalability of our Machine Learning Algorithm and for exploiting sparsity in our datasets. Knowledge of data structures (Binary Trees, Hashing, Heap, Stack etc), Dynamic Programming, Randomized & Sublinear Algorithm, Graphs, Gradient/Stochastic Descents and Primal-Dual methods are needed.
8. This comprises of other Math topics not covered in the four major areas described above. They include Real and Complex Analysis (Sets and Sequences, Topology, Metric Spaces, Single-Valued and Continuous Functions, Limits, Cauchy Kernel, Fourier Transforms), Information Theory (Entropy, Information Gain), Function Spaces and Manifolds.

Sys-

Python Control Systems Library (python-control) is a Python package that implements basic operations for analysis and design of feedback control systems.

Features

* Linear input/output systems in state-space and frequency domain
* Block diagram algebra: serial, parallel, and feedback interconnections
* Time response: initial, step, impulse
* Frequency response: Bode and Nyquist plots
* Control analysis: stability, reachability, observability, stability margins
* Control design: eigenvalue placement, LQR, H2, Hinf
* Model reduction: balanced realizations, Hankel singular values
* Estimator design: linear quadratic estimator (Kalman filter)

The value of **sys.argv** is a Python list of command line arguments that were passed to the Python script. The first argument, **argv[0]** is the name of the Python script itself. Depending on the platform that you are running on, the first argument may contain the full path to the script or just the file name.

>>> import sys

>>> sys.argv

['']

The value of **sys.executable** is the absolute path to the Python interpreter. This is useful when you are using someone else’s machine and need to know where Python is installed. On some systems, this command will fail and it will return an empty string or None.

>>> import sys

>>> sys.executable

'C:\\Python27\\pythonw.exe'

The **sys.exit()** function allows the developer to exit from Python. The **exit** function takes an optional argument, typically an integer, that gives an exit status. Zero is considered a “successful termination”. Be sure to check if your operating system has any special meanings for its exit statuses so that you can follow them in your own application. Note that when you call **exit**, it will raise the **SystemExit** exception, which allows cleanup functions to work in the **finally** clauses of **try / except** blocks.

>>> import sys

>>> sys.exit(0)

Traceback (most recent call last):

File "<pyshell#5>", line 1, in <module>

sys.exit(0)

SystemExit: 0

The sys module’s **path** value is a list of strings that specifies the search path for modules. Basically this tells Python what locations to look in when it tries to import a module. According to the Python documentation, **sys.path** is initialized from an environment variable called PYTHONPATH, plus an installation-dependent default.

>>> import sys

>>> print(sys.path)

['',

'C:\\Python27\\Lib\\idlelib',

'C:\\Python27\\lib\\site-packages\\setuptools-0.9.5-py2.7.egg',

'C:\\Python27\\lib\\site-packages\\pip-1.3.1-py2.7.egg',

'C:\\Python27\\lib\\site-packages\\sphinx-1.2b3-py2.7.egg',

'C:\\Python27\\lib\\site-packages\\docutils-0.11-py2.7.egg',

'C:\\Python27\\lib\\site-packages\\pygments-1.6-py2.7.egg',

'C:\\Windows\\system32\\python27.zip', '

C:\\Python27\\DLLs',

'C:\\Python27\\lib',

'C:\\Python27\\lib\\plat-win',

'C:\\Python27\\lib\\lib-tk',

'C:\\Python27',

'C:\\Python27\\lib\\site-packages',

'C:\\Python27\\lib\\site-packages\\PIL',

'C:\\Python27\\lib\\site-packages\\wx-2.9.4-msw']

The **sys.platform** value is a platform identifier. You can use this to append platform specific modules to **sys.path**, import different modules depending on platform or run different pieces of code.

>>> os = sys.platform

>>> if os == "win32":

# use Window-related code here

import \_winreg

elif os.startswith('linux'):

# do something Linux specific

import subprocess

subprocess.Popen(["ls, -l"])

The **stdin**, **stdout** and **stderr** map to file objects that correspond to the interpreter’s standard input, output and error streams, respectively. **stdin** is used for all input given to the interpreter except for scripts whereas **stdout** is used for the output of **print** and **expression** statements. The primary reason I mention this is that you will sometimes need to redirect stdout or stderr or both to a file.

OpenCv (cv2)-

1. A set of training samples. Each training sample is a vector of values (in Computer Vision it's sometimes referred to as feature vector). Usually all the vectors have the same number of components (features); OpenCV ml module assumes that. Each feature can be ordered (i.e. its values are floating-point numbers that can be compared with each other and strictly ordered, i.e. sorted) or categorical (i.e. its value belongs to a fixed set of values that can be integers, strings etc.).
2. Optional set of responses corresponding to the samples. Training data with no responses is used in unsupervised learning algorithms that learn structure of the supplied data based on distances between different samples. Training data with responses is used in supervised learning algorithms, which learn the function mapping samples to responses. Usually the responses are scalar values, ordered (when we deal with regression problem) or categorical (when we deal with classification problem; in this case the responses are often called "labels"). Some algorithms, most noticeably Neural networks, can handle not only scalar, but also multi-dimensional or vector responses.
3. Another optional component is the mask of missing measurements. Most algorithms require all the components in all the training samples be valid, but some other algorithms, such as decision tress, can handle the cases of missing measurements.
4. In the case of classification problem user may want to give different weights to different classes. This is useful, for example, when:
   * user wants to shift prediction accuracy towards lower false-alarm rate or higher hit-rate.
   * user wants to compensate for significantly different amounts of training samples from different classes.
5. In addition to that, each training sample may be given a weight, if user wants the algorithm to pay special attention to certain training samples and adjust the training model accordingly.
6. Also, user may wish not to use the whole training data at once, but rather use parts of it, e.g. to do parameter optimization via cross-validation procedure.

Numpy-

1. Numpy (Numerical Python) is a linear algebra library in Python. It is a very important library on which almost every data science or machine learning Python packages such as SciPy (Scientific Python), Mat−plotlib (plotting library), Scikit-learn, etc depends on to a reasonable extent.
2. NumPy is very useful for performing mathematical and logical operations on Arrays. It provides an abundance of useful features for operations on n-arrays and matrices in Python.
3. A NumPy array is simply a grid that contains values of the same type. NumPy Arrays come in two forms; Vectors and Matrices. Vectors are strictly one-dimensional(1-d) arrays, while Matrices are multidimensional. In some cases, Matrices can still have only one row or one column.
4. Numpy does broadcasting of arrays of different shapes during arithmetic operations. What this means in general is that the smaller array (or scalar) is broadcasted across the larger array so that they have compatible shapes. The example below shows an instance of broadcasting:

Collections –

1. Counter is a dict subclass which helps to count hashable objects. Inside it elements are stored as dictionary keys and counts are stored as values which can be zero or negative.
2. We have already encountered some simple Python types like numbers, strings and booleans. Now we will see how we can group multiple values together in a collection – like a list of numbers, or a dictionary which we can use to store and retrieve key-value pairs. Many useful collections are built-in types in Python, and we will encounter them quite often.

Pickle –

It is used for serializing and de-serializing a Python object structure. Any object in python can be pickled so that it can be saved on disk. What pickle does is that it “serialises” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream.

The module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. Pickling (and un pickling) is alternatively known as “serialization”, “marshalling,” or “flattening”, however, to avoid confusion, the terms used here are “pickling” and “unpickling”.

The pickle module has an optimized cousin called the cpickle module. As its name implies, cpickle is written in C, so it can be up to 1000 times faster than cpickle. However it does not support subclassing of the Pickler and unpickle classes, because in pickle these are functions, not classes. Most applications have no need for this functionality, and can benefit from the improved performance of cpickle. Other than that, the interfaces of the two modules are nearly identical; the common interface is described in this manual and differences are pointed out where necessary. In the following discussions, we use the term “pickle” to collectively describe the pickle and cpickle modules.

Python has a more primitive serialization module called marshal, but in general pickle should always be the preferred way to serialize Python objects. Marshal exists primarily to support Python’s .pyc files.

The pickle module differs from marshal in several significant ways:

* The pickle module keeps track of the objects it has already serialized, so that later references to the same object won’t be serialized again. Marshal doesn’t do this.

This has implications both for recursive objects and object sharing. Recursive objects are objects that contain references to themselves. These are not handled by marshal, and in fact, attempting to marshal recursive objects will crash your Python interpreter. Object sharing happens when there are multiple references to the same object in different places in the object hierarchy being serialized. pickle stores such objects only once, and ensures that all other references point to the master copy. Shared objects remain shared, which can be very important for mutable objects. pickle cannot be used to serialize user-defined classes and their instances. pickle can save and restore class instances transparently, however the class definition must be importable and live in the same module as when the object was stored.

* The pickle serialization format is not guaranteed to be portable across Python versions. Because its primary job in life is to support .pyc files, the Python implementers reserve the right to change the serialization format in non-backwards compatible ways should the need arise. The pickle serialization format is guaranteed to be backwards compatible across Python releases.

The data format used by pickle is Python-specific. This has the advantage that there are no restrictions imposed by external standards such as XDR (which can’t represent pointer sharing); however it means that non-Python programs may not be able to reconstruct pickled Python objects.

By default, the pickle data format uses a printable ASCII representation. This is slightly more voluminous than a binary representation. The big advantage of using printable ASCII (and of some other characteristics of pickle representation) is that for debugging or recovery purposes it is possible for a human to read the pickled file with a standard text editor.

There are currently 3 different protocols which can be used for pickling.

* Protocol version 0 is the original ASCII protocol and is backwards compatible with earlier versions of Python.
* Protocol version 1 is the old binary format which is also compatible with earlier versions of Python.
* Protocol version 2 was introduced in Python 2.3. It provides much more efficient pickling of new style classes.

Anaconda-

Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager, called Anaconda Navigator , so it eliminates the need to learn to install each library independently.

The open source packages can be individually installed from the Anaconda repository with the **conda install** command or using the **pip install** command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together.

Custom packages can be made using the **conda build** command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

### Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOSand Linux.

The following applications are available by default in Navigator [12]:

* JupyterLab
* Jupyter Notebook
* QtConsole
* Spyder
* Glueviz
* Orange
* Rstudio
* Visual Studio Code

### Conda

Main article: Conda (package manager)

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies.It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

## Anaconda Cloud

Anaconda Cloud is a package management service by Anaconda where you can find, access, store and share public and private notebooks, environments, and conda and PyPI packages. Cloud hosts useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in or to have a Cloud account, to search for public packages, download and install them.

**Hardware Requirements:**

System should have webcam and a good processor.

# 

# CHAPTER 2

## Product Backlog

As described in the [Scrum Guide](https://www.scrumguides.org/), the Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The [Product Owner](https://www.scrum.org/resources/what-is-a-product-owner) is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is never complete. The earliest development of it lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. If a product exists, its Product Backlog also exists.

The flow of project is converted into the form of product backlog to make the process easier into divided tasks and estimated time. In product backlog we specify the tasks in the form of user stories.

These user stories show that what the project is going to do. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the [Development Team](https://www.scrum.org/resources/what-is-a-scrum-development-team) collaborate on the details of Product Backlog items.

During Product Backlog refinement, items are reviewed and revised. Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product.

A Product Backlog attribute that groups items may then be employed.

The project is divided into 4 phases. In these 4 Phases the project will be completed. In product backlog the tasks are divided and user stories are made to understand the flow and role of users in this projects. In each sprint backlog each task is allotted to a member.

There are 3 types of users in this project-

* Developer
* User
* Database admin

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PRODUCT BACKLOG** | | | | | | | | |
|  |  |  | **Batch 2018\_2019** |  |  |  |  |  |
| **SPRINT BACKLOG** | **USER ID** | **BACKLOG ITEM** | | | **PRIORITY** | **RESPONSIBLE** | **ESTIMATED DATE** | **REMARKS** |
| **AS A/AN** | **I WANT TO** | **SO THAT** |
| 1 | SB1/US1 | Developer | Understand the problem | I can search for relevant solutions | 1 |  |  |  |
| 1 | SB1/US2 | Developer | Understand user stories | I can search for dataset accordingly | 1 |  |  |  |
| 1 | SB1/US3 | User | Set object to write | I can set the object | 1 |  |  |  |
| 1 | SB1/US4 | User | Save the object data locally | I don’t have to set object each time |  |  |  |  |
| 1 | SB1/US5 | User | Change the object | I can change the writing object | 1 |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 2 | SB2/US1 | User | Write on the screen without physical interaction | I can write on the screens | 2 |  |  |  |
| 2 | SB2/US2 | Developer | Trace the object user using to write | I can get the signature on the screen | 2 |  |  |  |
| 2 | SB2/US3 | Developer | Save the sign into folder | Signature will be saved | 2 |  |  |  |
| 2 | SB2/US5 | User | Retry | I can try again if my previous signature is wrong | 3 |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 3 | SB3/US1 | Developer | Find datasets | I can use it | 3 |  |  |  |
| 3 | SB3/US2 | Developer | Download datasets from various sources | I can train model on it | 3 |  |  |  |
| 3 | SB3/US3 | Developer | Normalize and set data | I can simplify my dataset | 3 |  |  |  |
| 3 | SB3/US4 | Developer | Train different ML models on standard dataset | I can test it's accuracy and performance on each algorithm | 3 |  |  |  |
| 3 | SB3/US5 | Developer | Train different DL models on standard dataset | I can test it's accuracy and performance on each algorithm | 3 |  |  |  |
| 3 | SB3/US6 | Developer | Train different ML models on my dataset | I can test it's accuracy and performance on each algorithm | 3 |  |  |  |
| 3 | SB3/US7 | Developer | Train different DL models on my dataset | I can test it's accuracy and performance on each algorithm | 3 |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 4 | SB4/US1 | Developer | Choose most accurate algorithm | I can implement that and get best results |  |  |  |  |
| 4 | SB4/US2 | Developer | Improve the input | I can improve the model | 4 |  |  |  |
| 4 | SB4/US3 | Developer | Apply my trained model on user stories | Application can be used | 4 |  |  |  |
| 4 | SB4/US4 | Developer | Improve the model | Accuracy of the model will be improved | 4 |  |  |  |
| 4 | SB4/US5 | Developer | Implement the final model for general user | Application can be used by general user | 4 |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## 

**Sprint Backlog-1**

## In this phase all the perquisites for the project are gathered.

## Search for other relevant solutions for the project analysing the similar projects.

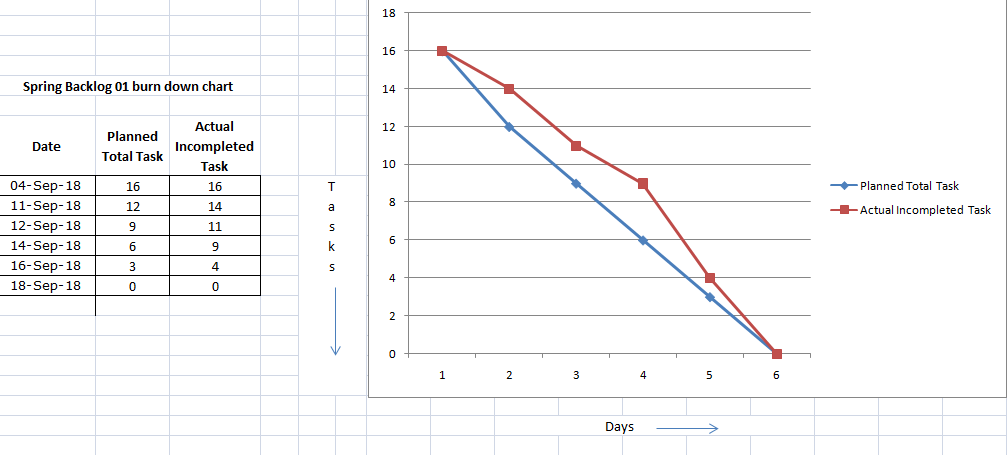
## Then a web form is developed to take the details of user.

## Design a database schema for the project and create databases to store the data of each user. Read about the concepts and methods which will be used in project for example Opencv.

## 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SPRINT BACKLOG 1** | | | | | | |
|  |  |  |  |  |  |  |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** | **ESTIMATED DATE OF TASK COMPLETION** |
|  |  |  |  |  |  |  |
| **SPRINT 1 - <Signature authentication in Air>** | | | | | | |
| SB1/US1 | Understand the problem | SB1/D1/T1 | Breaking the problem and understanding each component | AS+AK+LC | COMPLETED |  |
| SB1/D1/T2 | Searching for relavent solutions | AS+AK+LC | COMPLETED |  |
| SB1/D1/T3 | Searching alternative approches | AS+AK+LC | COMPLETED |  |
| SB1/D1/T4 | Analysing similar projects | AS+AK+LC | COMPLETED |  |
| SB1/US2 | Understand user stories | SB1/D2/T1 | Looking for various use cases | AS+AK+LC | COMPLETED |  |
| SB1/D2/T2 | Discussion about user stories | AS+AK+LC | COMPLETED |  |
| SB1/D2/T3 | Searching for type of data required | AS+AK+LC | COMPLETED |  |
| SB1/US3 | Set object to write | SB1/D3/T1 | Set the object constraints | AK | COMPLETED |  |
| SB1/D3/T2 | Set the object color | AS+AK+LC | COMPLETED |  |
| SB1/D3/T3 | Set HSV values | LC | COMPLETED |  |
| SB1/US4 | Save the object data locally | SB1/D4/T1 | Save the constraints of object | AS+AK+LC | COMPLETED |  |
| SB1/D4/T2 | Save data in pickle | AS+AK+LC | COMPLETED |  |
| SB1/D4/T3 | Load data from pickle | AS | COMPLETED |  |
| SB1/US5 | Change the object | SB1/D5/T1 | Change the object constraints | AS | COMPLETED |  |
| SB1/D5/T2 | Save the constraints of changed object data | AS | COMPLETED |  |
|  |  |  |  |  |  |  |

## Burn Down Chart-1



## Fig. 1

**Output of Spring Backlog 1**

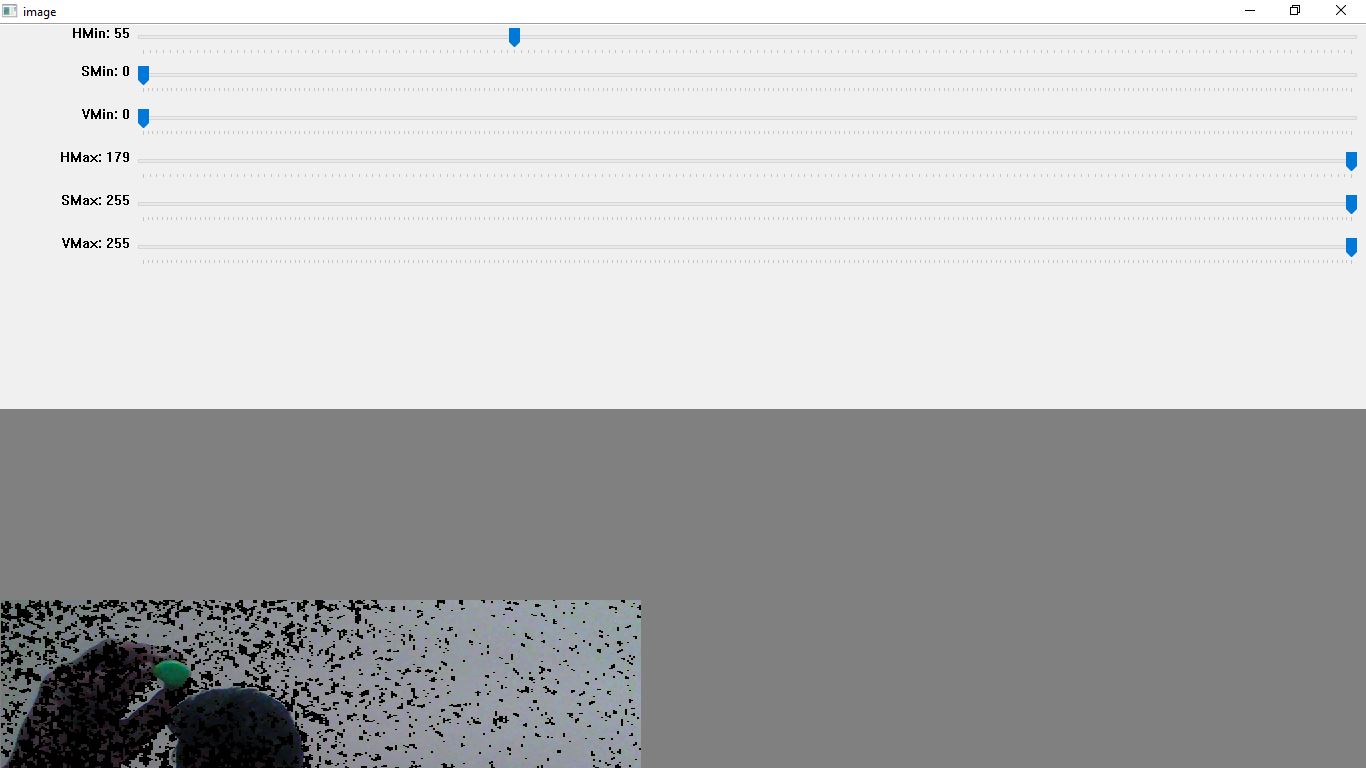
****

Fig. 2

## Sprint Backlog-2

* In this phase we developed the application where user can write using the webcam.
* You don’t need any physical device for writing on screen simply write on the air
* Store the data which is written on the screen and any modification can be done or data

can be deleted if it will not require.

* The burn down chart shows the actual task and the planned tasks.
* User can modifies his details.

**Spring Backlog-2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SPRINT BACKLOG 2** | | | | | | |
|  |  |  |  |  |  |  |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** | **ESTIMATED DATE OF TASK COMPLETION** |
|  |  |  |  |  |  |  |
| **SPRINT 2 - <Signature authentication in Air>** | | | | | | |
| SB2/US1 | Write on the screen without physical interaction | SB2/D1/T1 | Open the Webcam | AS+AK+LC | COMPLETED |  |
| SB2/D1/T2 | Write in front of the Camera | AS+AK+LC | COMPLETED |  |
| SB2/D1/T3 | Add Validations | AS+AK+LC | COMPLETED |  |
| SB2/D1/T4 | Store the data | AS+AK+LC | COMPLETED |  |
| SB2/US2 | Trace the object | SB2/D2/T1 | Trace the object on screen | AS+AK+LC | COMPLETED |  |
| SB2/D2/T2 | Convert B&W image | AS+AK+LC | COMPLETED |  |
| SB2/D2/T3 | Save the Data | AS+AK+LC | COMPLETED |  |
| SB2/US3 | Save the sign into folder | SB2/D3/T1 | Create a folder to store | AS+AK+LC | COMPLETED |  |
| SB2/D3/T2 | Save the Data in that folder | AS+AK+LC | COMPLETED |  |
| SB2/US4 | Retry | SB2/D5/T1 | Again try different sign | AS+AK+LC | COMPLETED |  |
| SB2/D5/T2 | Save the data | AS+AK+LC | COMPLETED |  |
|  |  |  |  |  |  |  |

**Burn Down Chart-2**

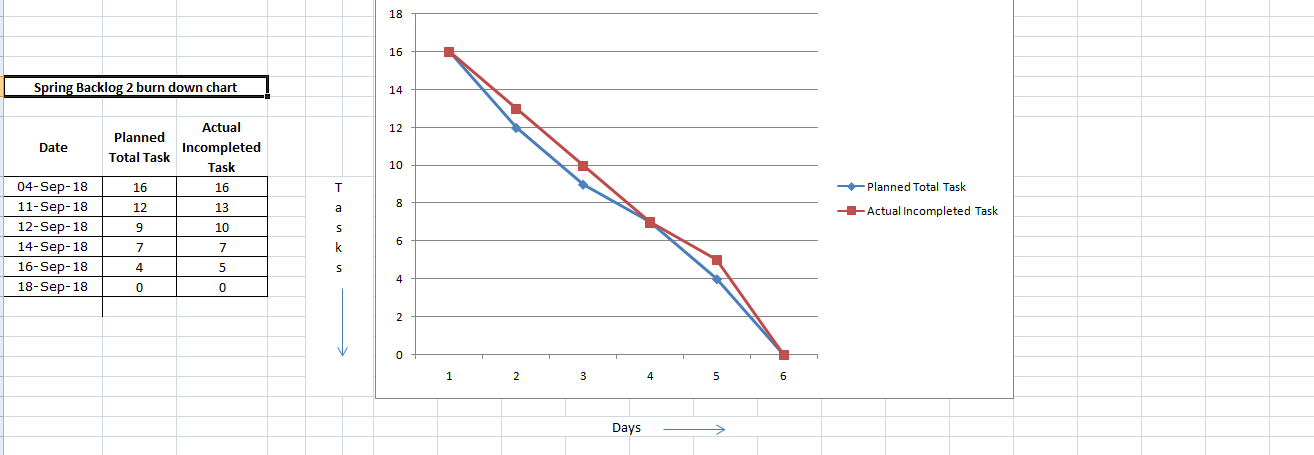


Fig.3

**Output of Spring Backlog 2**



Fig.4**Sprint Backlog-3**

* In this phase we download the datasets from various websites (kaggle.com,github) and this dataset for the project.
* This dataset apply on various data models and check the accuracy.
* The model chooses the higher accuracy and applies on the data set which is downloaded from the websites.
* Data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0.
* Normalization is a good technique to use when you do not know the distribution of your data or when you know the distribution is not Gaussian (a bell curve).
* We apply different DL models on Standard Datasets and store the results.
* The figure shows the complete sprint backlog 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SPRINT BACKLOG 3** | | | | | | |
|  |  |  |  |  |  |  |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** | **ESTIMATED DATE OF TASK COMPLETION** |
|  |  |  |  |  |  |  |
| **SPRINT 3 - <Signature authentication in Air>** | | | | | | |
| SB3/US1 | Find datasets | SB3/D1/T1 | Find the data set from different sources | AS+AK+LC | COMPLETED |  |
| SB3/D1/T2 | Select relavent data set | AS+AK+LC | COMPLETED |  |
| SB3/US2 | Download datasets from various sources | SB3/D2/T1 | Download datasets Github | AS | COMPLETED |  |
| SB3/D2/T2 | Download datasets kaggle | LC | COMPLETED |  |
| SB3/D2/T3 | Other Sources | AK | COMPLETED |  |
| SB3/US3 | Normalize and set the data | SB3/D3/T1 | Normalize the data | AS+AK+LC | COMPLETED |  |
| SB3/D3/T2 | Set the data | AS+AK+LC | COMPLETED |  |
| SB3/US4 | Train different ML models on standard dataset | SB3/D4/T1 | Apply ML models on Standard DataSet | AS+AK+LC | COMPLETED |  |
| SB3/D4/T2 | Store result | AS+AK+LC | COMPLETED |  |
| SB3/US5 | Train different DL models on standard dataset | SB3/D4/T1 | Apply different DL models on Standard DataSet | AS+AK+LC | COMPLETED |  |
| SB3/D4/T2 | Store result | AS+AK+LC | COMPLETED |  |
| SB3/US6 | Train different ML models on my own dataset | SB3/D4/T1 | Apply ML models on my own dataset | AS+AK+LC | COMPLETED |  |
| SB3/D4/T2 | Store result | AS+AK+LC | COMPLETED |  |
| SB3/US7 | Train different DL models on my own dataset | SB3/D4/T1 | Apply different DL models on my own dataset | AS+AK+LC | COMPLETED |  |
| SB3/D4/T2 | Store result | AS+AK+LC | COMPLETED |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Burn down chart -3**

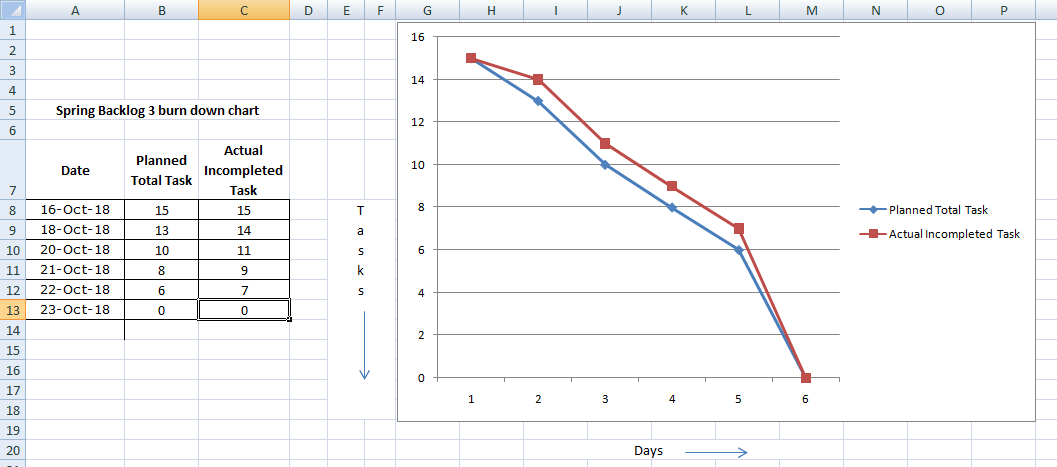
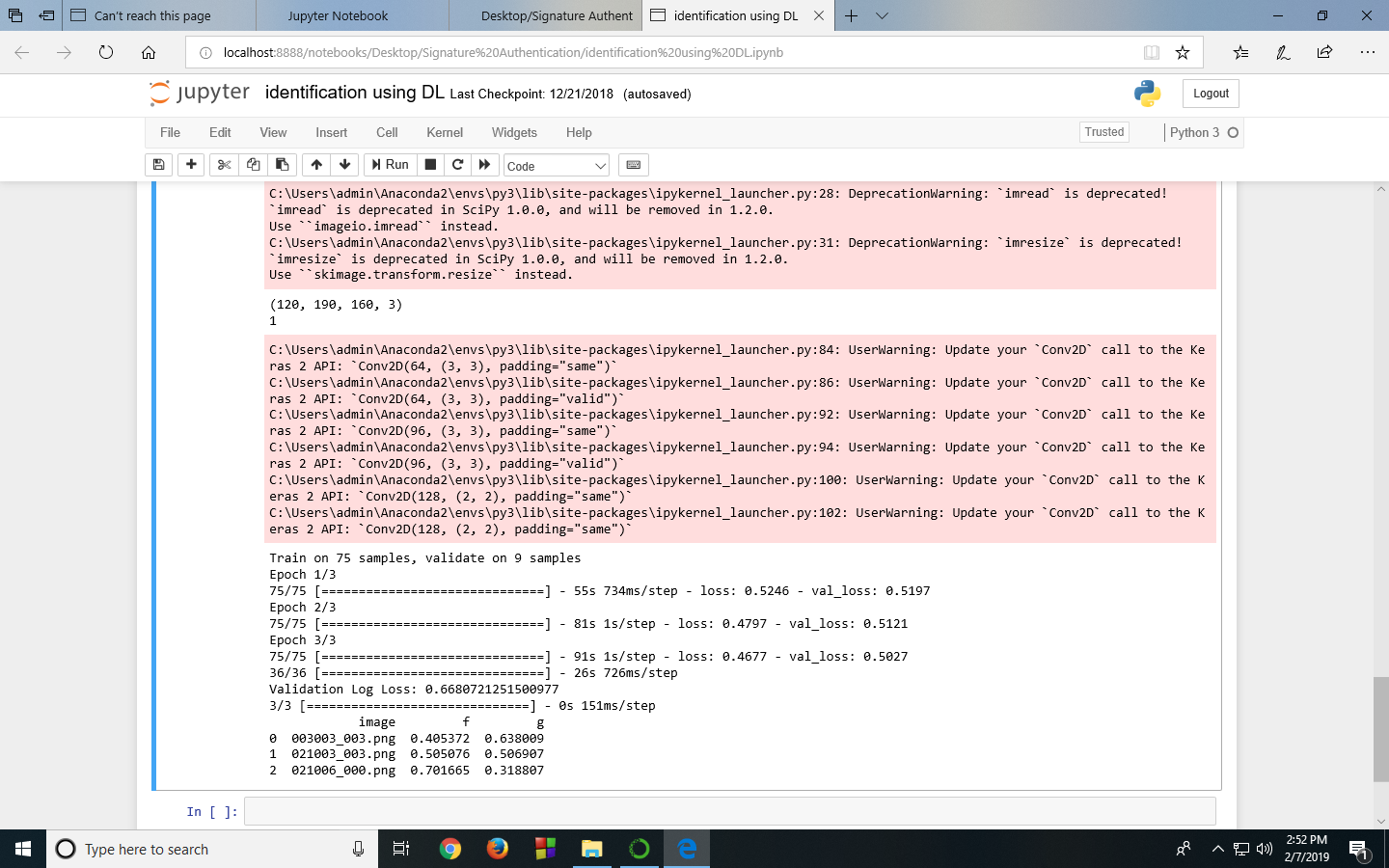
****

Fig. 5

**Output of Spring Backlog 3**



## Fig.6

## Sprint Backlog-4

* In this phase we compare all the data models and choose the data model which is having higher accuracy.
* We make own datasets and apply on the data models and see the accuracy is same as on downloaded dataset.
* For same accuracy we normalize the own datasets by transforming and rescaling the data.
* Check the status of data model and improved the dataset for more accurate results.

And finally implement it for all users.

* Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.
* The figure shows the sprint backlog 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SPRINT BACKLOG 4** | | | | | | |
|  |  |  |  |  |  |  |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** | **ESTIMATED DATE OF TASK COMPLETION** |
|  |  |  |  |  |  |  |
| **SPRINT 4 - <Signature authentication in Air>** | | | | | | |
| SB4/US1 | Choose most accurate algorithm | SB4/D1/T1 | Compare all the ML models | AS+AK+LC | COMPLETED |  |
| SB4/D1/T2 | Choice best efficient model | AS+AK+LC | COMPLETED |  |
| SB4/D1/T3 | Apply dataset on efficient model | AS+AK+LC | COMPLETED |  |
| SB4/D1/T4 | Check accuracy | AS+AK+LC | COMPLETED |  |
| SB4/US2 | Improve the input | SB4/D2/T1 | Change the dataset and model | AS+AK+LC | COMPLETED |  |
| SB4/D2/T2 | Check accuracy | AS+AK+LC | COMPLETED |  |
| SB4/US3 | Apply my trained model on user stories | SB4/D3/T1 | Apply my trained model on user stories | AS+AK+LC | COMPLETED |  |
| SB4/D3/T2 | Check model status | AS+AK+LC | COMPLETED |  |
| SB4/US4 | Improve the model | SB4/D4/T1 | Improve the model with more accurate data | AS+AK+LC | COMPLETED |  |
| SB4/US5 | Implement the final model for general user | SB4/D5/T1 | Implement the model for all the users | AS+AK+LC | COMPLETED |  |
|  |  |  |  |  |  |  |

**Burn Down Chart-4**

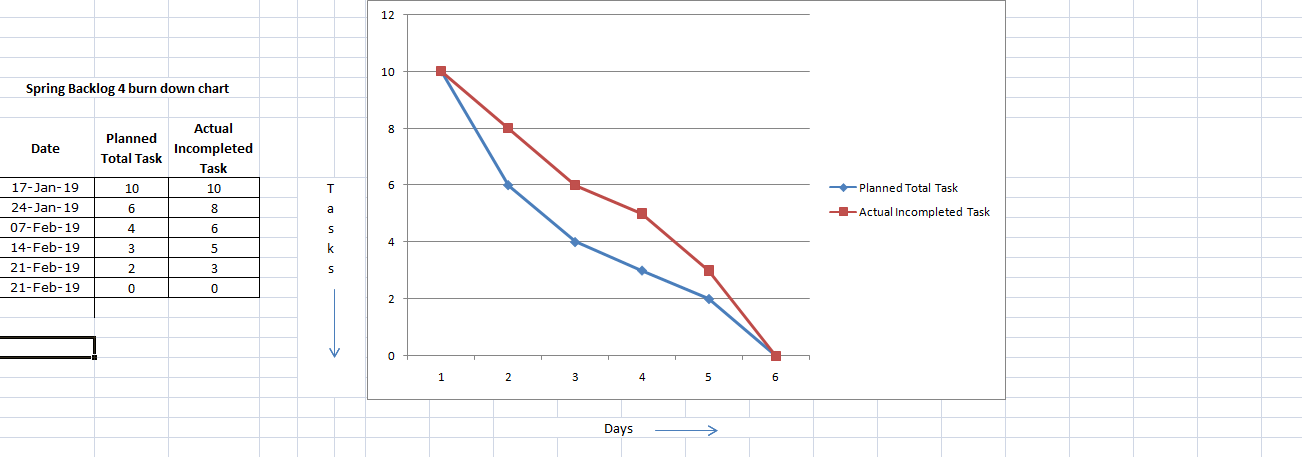


Fig. 7

**CHAPTER 3**

# TECHNOLOGY APPLIED AND PROJECT MANAGEMENT

## Project Management :

Project management is the application of processes, methods, knowledge, skills and experience to achieve the project objectives. General . A project is a unique, transient endeavor, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

Project management is the practice of initiating, planning, executing, controlling, and closing the [work](https://en.wikipedia.org/wiki/Work_(project_management)) of a [team](https://en.wikipedia.org/wiki/Project_team) to achieve specific goals and meet specific success criteria at the specified time. A [project](https://en.wikipedia.org/wiki/Project) is a temporary endeavor designed to produce a unique product, service or result with a defined beginning and end undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with [business as usual](https://en.wikipedia.org/wiki/Business_operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the [management](https://en.wikipedia.org/wiki/Management) of such distinct production approaches requires the development of distinct technical skills and management strategies.

## Software Project Management

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of [project management](https://en.wikipedia.org/wiki/Project_management) in which [software](https://en.wikipedia.org/wiki/Software) projects are planned, implemented, monitored and controlled.

The job pattern of an IT company engaged in software development can be seen split in two parts:

* Software Creation
* Software Project Management

A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). A Project can be characterized as:

* Every project may have a unique and distinct goal.
* Project is not routine activity or day-to-day operations.
* Project comes with a start time and end time.
* Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
* Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

## Software Project

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

## Need of software project management

Software is said to be an intangible product. Software development is a kind of all new stream in world business and there’s very little experience in building software products. Most software products are tailor made to fit client’s requirements. The most important is that the underlying technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one. All such business and environmental constraints bring risk in software development hence it is essential to manage software projects efficiently.



The image above shows triple constraints for software projects. It is an essential part of software organization to deliver quality product, keeping the cost within client’s budget constrain and deliver the project as per scheduled. There are several factors, both internal and external, which may impact this triple constrain triangle. Any of three factor can severely impact the other two.

Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

## 

## Software Project Manager

A software project manager is a person who undertakes the responsibility of executing the software project. Software project manager is thoroughly aware of all the phases of SDLC that the software would go through. Project manager may never directly involve in producing the end product but he controls and manages the activities involved in production.

A project manager closely monitors the development process, prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.

Let us see few responsibilities that a project manager shoulders -

**Managing People**

* Act as project leader
* Liaison with stakeholders
* Managing human resources
* Setting up reporting hierarchy etc.

**Managing Project**

* Defining and setting up project scope
* Managing project management activities
* Monitoring progress and performance
* Risk analysis at every phase
* Take necessary step to avoid or come out of problems
* Act as project spokesperson

## Software Management Activities

Software project management comprises of a number of activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. Project management activities may include:

* **Project Planning**
* **Scope Management**
* **Project Estimation**

## 

## Project Planning

Software project planning is task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production. Project planning may include the following:

## Scope Management

It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product. Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun.

During Project Scope management, it is necessary to -

* Define the scope
* Decide its verification and control
* Divide the project into various smaller parts for ease of management.
* Verify the scope
* Control the scope by incorporating changes to the scope

## Project Estimation

For an effective management accurate estimation of various measures is a must. With correct estimation managers can manage and control the project more efficiently and effectively.

Project estimation may involve the following:

* **Software size estimation**

Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software. Lines of code depend upon coding practices and Function points vary according to the user or software requirement.

* **Effort estimation**

The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers’ experience, organization’s historical data or software size can be converted into efforts by using some standard formulae.

* **Time estimation**

Once size and efforts are estimated, the time required to produce the software can be estimated. An effort required is segregated into sub categories as per the requirement specifications and interdependency of various components of software. Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS). The tasks are scheduled on day-to-day basis or in calendar months.

The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

* **Cost estimation**

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider -

* + Size of software
  + Software quality
  + Hardware
  + Additional software or tools, licenses etc.
  + Skilled personnel with task-specific skills
  + Travel involved
  + Communication

## Project Estimation Techniques

We discussed various parameters involving project estimation such as size, effort, time and cost. Project manager can estimate the listed factors using two broadly recognized techniques

## Decomposition Technique

This technique assumes the software as a product of various compositions.

There are two main models -

* **Line of Code** Estimation is done on behalf of number of line of codes in the software product.
* **Function Points** Estimation is done on behalf of number of function points in the software product.

## Empirical Estimation Technique

This technique uses empirically derived formulae to make estimation.These formulae are based on LOC or FPs.

* **Putnam Model**

This model is made by Lawrence H. Putnam, which is based on Norden’s frequency distribution (Rayleigh curve). Putnam model maps time and efforts required with software size.

* **COCOMO**

COCOMO stands for COnstructiveCOstMOdel, developed by Barry W. Boehm. It divides the software product into three categories of software: organic, semi-detached and embedded.

## Project Scheduling

Project Scheduling in a project refers to roadmap of all activities to be done with specified order and within time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and they arrange them keeping various factors in mind. They look for tasks lie in critical path in the schedule, which are necessary to complete in specific manner and strictly within the time allocated. Arrangement of tasks which lies out of critical path are less likely to impact over all schedule of the project.

For scheduling a project, it is necessary to -

* Break down the project tasks into smaller, manageable form
* Find out various tasks and correlate them
* Estimate time frame required for each task
* Divide time into work-units
* Assign adequate number of work-units for each task
* Calculate total time required for the project from start to finish

**Resource Management**

All elements used to develop a software product may be assumed as resource for that project. This may include human resource, productive tools and software libraries.

The resources are available in limited quantity and stay in the organization as a pool of assets. The shortage of resources hampers the development of project and it can lag behind the schedule. Allocating extra resources increases development cost in the end. It is therefore necessary to estimate and allocate adequate resources for the project.

Resource management includes -

* Defining proper organization project by creating a project team and allocating responsibilities to each team member
* Determining resources required at a particular stage and their availability
* Manage Resources by generating resource request when they are required and de-allocating them when they are no more needed.

## Project Risk Management

Risk management involves all activities pertaining to identification, analysing and making provision for predictable and non-predictable risks in the project. Risk may include the following:

* Experienced staff leaving the project and new staff coming in.
* Change in organizational management.
* Requirement change or misinterpreting requirement.
* Under-estimation of required time and resources.
* Technological changes, environmental changes, business competition.

## Risk Management Process

There are following activities involved in risk management process:

* **Identification -** Make note of all possible risks, which may occur in the project.
* **Categorize -** Categorize known risks into high, medium and low risk intensity as per their possible impact on the project.
* **Manage -** Analyze the probability of occurrence of risks at various phases. Make plan to avoid or face risks. Attempt to minimize their side-effects.
* **Monitor -** Closely monitor the potential risks and their early symptoms. Also monitor the effects of steps taken to mitigate or avoid them.

**Project Execution & Monitoring**

In this phase, the tasks described in project plans are executed according to their schedules.

Execution needs monitoring in order to check whether everything is going according to the plan. Monitoring is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.

These measures include -

* **Activity Monitoring -** All activities scheduled within some task can be monitored on day-to-day basis. When all activities in a task are completed, it is considered as complete.
* **Status Reports -** The reports contain status of activities and tasks completed within a given time frame, generally a week. Status can be marked as finished, pending or work-in-progress etc.
* **Milestones Checklist -** Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC. This milestone checklist is prepared once every few weeks and reports the status of milestones.

## 

## Project Communication Management

Effective communication plays vital role in the success of a project. It bridges gaps between client and the organization, among the team members as well as other stake holders in the project such as hardware suppliers.

Communication can be oral or written. Communication management process may have the following steps:

* **Planning** - This step includes the identifications of all the stakeholders in the project and the mode of communication among them. It also considers if any additional communication facilities are required.
* **Sharing** - After determining various aspects of planning, manager focuses on sharing correct information with the correct person on correct time. This keeps every one involved the project up to date with project progress and its status.
* **Feedback** - Project managers use various measures and feedback mechanism and create status and performance reports. This mechanism ensures that input from various stakeholders is coming to the project manager as their feedback.
* **Closure** - At the end of each major event, end of a phase of SDLC or end of the project itself, administrative closure is formally announced to update every stakeholder by sending email, by distributing a hardcopy of document or by other mean of effective communication.

After closure, the team moves to next phase or project.

## Configuration Management

Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions and development of the product.

IEEE defines it as “the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”.

Generally, once the SRS is finalized there is less chance of requirement of changes from user. If they occur, the changes are addressed only with prior approval of higher management, as there is a possibility of cost and time overrun.

## Project Management Tools:

Project management required tools to manage the work , time and resources. At present many of the software are available for project management. Some of the popular software tools are as follows.

### 01. [Trello](http://send.getapp.com/aff_c?offer_id=677&aff_id=1371)

Trello is an project management tool, instead this app is a free visual way to to glance at the entire project with a single view. With Trello you can organise cards, these cards can be your thoughts, conversations and to-do lists and be placed on a board for everyone to collaborate on.

### 02. [Basecamp](http://send.getapp.com/aff_c?offer_id=637&aff_id=1371)

Basecamp is the granddaddy of project management apps. Basecamp is considered the leading project management tool around. It boost a simple and easy to use interface to collaborate with your team and client. It allows you to create multiple projects and setup discussions, write to-do lists, manage files, create and share documents, and organise dates for scheduling.

### 03. [Teamwork Projects](http://send.getapp.com/aff_c?offer_id=947&aff_id=1371)

Teamwork Projects is the ultimate productivity tool to manage projects with your team. Teamwork allows you to keep all your projects, tasks and files all in one place and easily collaborate with a team. Teamwork helps you to visualise the entire project through a marked calendar and gantt chart and setup reporting. Teamwork supports file management with Google Drive, Box.com and Dropbox. As well as integration with leading apps such as third party accounting software and customer support apps.

### 04. [Resource Guru](https://resourceguruapp.com/)

Billed as the "simple way to schedule people, equipment and other resources", Resource Guru is a streamlined resource scheduling and leave management tool that’s designed to keep your projects on track. You can plan your team's workloads, receive daily booking reminders, report on KPIs, and more. Apple, Saatchi & Saatchi and Deloitte are among some of the cloud-based team calendar’s heavyweight customers.

### 05. [ActiveCollab](http://send.getapp.com/aff_c?offer_id=949&aff_id=1371)

ActiveCollab recently released its new version 5.0. The new revamped app is now more powerful and focused project management tool. It offers team collaborating features, task management, time tracking and importing expenses. One of the biggest asset of ActiveCollab is it offers invoicing features. You are able to track payments and expenses and have invoices paid directly within ActiveCollab with PayPal, and other credit card payments.

### 06. [Zoho Projects](http://send.appdoubler.com/aff_c?offer_id=101&aff_id=1371)

Zoho offers a wide range of business software including Projects. Zoho Projects is an proficient tool to project plan and project coordinator from start to finish. It boost all the features you need for project management with some advance features including reporting, integration with Google Apps and Dropbox, bug tracking, setup Wiki Pages to build a repository of information, forums and more.

### 07. [Jira](http://send.getapp.com/aff_c?offer_id=281&aff_id=1371)

Jira is specifically targeted for software development teams. Jira offers abilities to raise issues and bugs. Jira makes it real easy to track bugs and see which issues are still outstanding and how much time was spent on each task. Jira offer other products including Confluence a document collaboration tool, and HipChat a team chat and video and file sharing platform and other products.

### 08. [Asana](http://send.getapp.com/aff_c?offer_id=587&aff_id=1371)

Asana is the easiest way for teams to track their work so everyone knows who's doing what, by when. With tasks, projects, conversations and dashboards, Asana keeps your work organized, and teammates accountable so you can move work forward faster. Asana also lets you keep track of your work wherever you are with mobile apps for both iOS and Android.

### 09. [Podio](http://send.getapp.com/aff_c?offer_id=951&aff_id=1371)

Podio is a ever growing tool to organise and communication tool for any business. Podio allows you to personalise this platform to fit your business needs. Besides being able to communicate with a team, setup task management, use as a file storage system, like a traditional project management app, Podio can be an internal intranet for all your colleagues and departments to interact.

### 10. [Freedcamp](https://freedcamp.com/)

Whatever your project may be, either setting up an event, a web project or organising a wedding, Freedcamp helps you organise and plan effectively. Freedcamp has an organised dashboard to view the entire project at a glance. You can easily setup tasks, use sticky notes to visually setup tasks and organise them into the calendar. Freedcamp provides advance add-ons for high level business use including CRM, invoicing, issue tracking and setting up wiki pages.

### 11. [Wrike](http://send.getapp.com/aff_c?offer_id=239&aff_id=1371)

Wrike is advance application to help you work smarter. By making sure you are always staying on track and ensure you have the adequate resources to finish on time and on budget.Setting up tasks, engage your team and integrate with your business tools including Google Apps, Microsoft Excel, Dropbox and many more is so easy with Wrike.

## PO and Their Relevance to project

**PO1: Engineering knowledge:**Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

In this project creation process engineering knowledge of the software engineering and Electronics engineering have been applied. we have used software engineering , HTML,xml, Java , Android , Java Script , Php , J2EE, Data Base , Oracle , my sql , mango and other programming language and database to the project. We have applied all above engineering subjects in our projects.

**PO2: Problem analysis:**Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

In our projects we have identified an problem , once verified by the client we have worked to identify the solution using all of our theoretical and practical knowledge.

**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:**Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

In the project development we have applied Integrated Development Environment IDE for the rapid development of the code, used web server for the software development.

**PO6: The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

In 1961 , the Conference of Engineering Societies of Western Europe and the United States of America defined "professional engineer" as follows.

A professional engineer is competent by virtue of his/her fundamental education and training to apply the scientific method and outlook to the analysis and solution of engineering problems. He/she is able to assume personal responsibility for the development and application of engineering science and knowledge, notably in research, design, construction, manufacturing, superintending, managing and in the education of the engineer. His/her work is predominantly intellectual and varied and not of a routine mental or physical character. It requires the exercise of original thought and judgement and the ability to supervise the technical and administrative work of others. His/her education will have been such as to make him/her capable of closely and continuously following progress in his/her branch of engineering science by consulting newly published works on a worldwide basis, assimilating such information and applying it independently. He/she is thus placed in a position to make contributions to the development of engineering science or its applications. His/her education and training will have been such that he/she will have acquired a broad and general appreciation of the engineering sciences as well as thorough insight into the special features of his/her own branch. In due time he/she will be able to give authoritative technical advice and to assume responsibility for the direction of important tasks in his/her branch.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Sustainability is the ability to continue a defined behavior indefinitely. Sometimes environmental, social and economic are termed to be the three pillars of sustainability.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. Engineers uphold and advance the integrity, honor and dignity of the engineering profession by:

1. Using their knowledge and skill for the enhancement of human welfare;
2. being honest and impartial, and servicing with fidelity the public, their employers and clients;
3. Striving to increase the competence and prestige of the engineering profession; and
4. Supporting the professional and technical societies of their disciplines.

**PO9. Individual and Team work**: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

To work successful in team a team member must have following capabilities.

**1. The Ability to Listen**

it is important to listen to one another's ideas. Too often in a business setting, you have a group of people simply waiting for their turn to speak, not paying one iota of attention to the persons on their left or right. So it is a good teamwork skill to have the ability to listen

**2. Check Your Ego**

This isn't saying abandon your ego all together, because that isn't healthy. But leaving your ego at the door temporarily is a very important team work skill. The reason this is so essential is because there is always someone better than you at something, no matter how brilliant you are.

**3. Critique**

By critique, I mean constructive criticism. Be able to give others constructive criticism and be able to listen to others critique your ideas and work. There shouldn't be any offense taken to constructive criticism. You all want to succeed, and this is a vital step in doing so.

**4. Delegation**

The mentality must be applied to teamwork. Delegate roles to those who do them best.

**5. Show Respect**

If you and another person happen to be paired up and can't stand each other, you can still put that aside for a couple of hours, treat each other civilly, and complete the tasks at hand. You may even overcome the dislike toward one another.

**6. Be Helpful**

This is simple.If one of your teammates does not understand an idea, discussion, or task that is being completed, take the necessary time to explain it to them and work with them. There are no weak links when everyone helps one another. Some take longer to learn than others, but that doesn't mean that they are of less intelligence. If in a meeting someone asks a question because they don't understand, don't frown at them. Just answer the questions patiently and concisely.

**7. Question One Another**

If someone brings up a topic of discussion and a solution to this topic, question them. Respectfully question, don't badger. Rather, ask them how it will work, why it will work over the long-run, and how everyone else can implement the idea.

**8. Participation**

Have the entire team encourage shy people to engage in the topics of discussion. Don't demand it, but make them realize that you really want to hear their ideas.

**9. Rational Debate**

Bad ideas are bad for teams. Spirited, friendly, rational debate is where facts come forward, ideas are born, and quality rises to the top.

**10. Set The Right Environment**

Try to make the space in which your team is assembled as comfortable, relaxing, and inviting as possible. You do not want your team to be tense and with frayed nerves.

**PO 10: Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project Management and Finance:**Demonstrate knowledge and understanding of the engineering management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Project management is the application of processes, methods, knowledge, skills and experience to achieve the project objectives. In general project is a unique, transient endeavour, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

**PO12: Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Life Long Learning means is the provision or use of both formal and informal learning opportunities throughout people's lives in order to foster the continuous development and improvement of the knowledge and skills needed for employment and personal fulfillment

# CHAPTER 4

# PROJECT IMPLEMENTATION

## Sprint Backlog-1

**Code for Set the Object**

import math

import sys

import cv2

importnumpy as np

import random

from collections import deque

import pickle

ch = int(input("\nDo you want to set the object first?\n\t1. Yes\n\t2. No\n\t"))

ocount = pickle.load(open("ocount.p", "rb"))

if(ch == 1):

def nothing(x):

pass

cv2.namedWindow('image')

cv2.createTrackbar('HMin','image',0,179,nothing)

cv2.createTrackbar('SMin','image',0,255,nothing)

cv2.createTrackbar('VMin','image',0,255,nothing)

cv2.createTrackbar('HMax','image',0,179,nothing)

cv2.createTrackbar('SMax','image',0,255,nothing)

cv2.createTrackbar('VMax','image',0,255,nothing)

cv2.setTrackbarPos('HMax', 'image', 179)

cv2.setTrackbarPos('SMax', 'image', 255)

cv2.setTrackbarPos('VMax', 'image', 255)

hMin = sMin = vMin = hMax = sMax = vMax = 0

phMin = psMin = pvMin = phMax = psMax = pvMax = 0

cap = cv2.VideoCapture(0)

waitTime = 330

while(1):

ret, img = cap.read()

output = img

hMin = cv2.getTrackbarPos('HMin','image')

sMin = cv2.getTrackbarPos('SMin','image')

vMin = cv2.getTrackbarPos('VMin','image')

hMax = cv2.getTrackbarPos('HMax','image')

sMax = cv2.getTrackbarPos('SMax','image')

vMax = cv2.getTrackbarPos('VMax','image')

lower = np.array([hMin, sMin, vMin])

upper = np.array([hMax, sMax, vMax])

hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, lower, upper)

output = cv2.bitwise\_and(img,img, mask= mask)

if( (phMin != hMin) | (psMin != sMin) | (pvMin != vMin) | (phMax != hMax) | (psMax != sMax) | (pvMax != vMax) ):

phMin = hMin

psMin = sMin

pvMin = vMin

phMax = hMax

psMax = sMax

pvMax = vMax

old\_values = {"hMin": hMin, "sMin": sMin, "vMin": vMin, "hMax": hMax, "sMax": sMax, "vMax": vMax}

pickle.dump(old\_values, open("old\_values.p", "wb"))

output = cv2.flip(output,1)

cv2.imshow('image',output)

k = cv2.waitKey(5) & 0xFF

if k == 27:

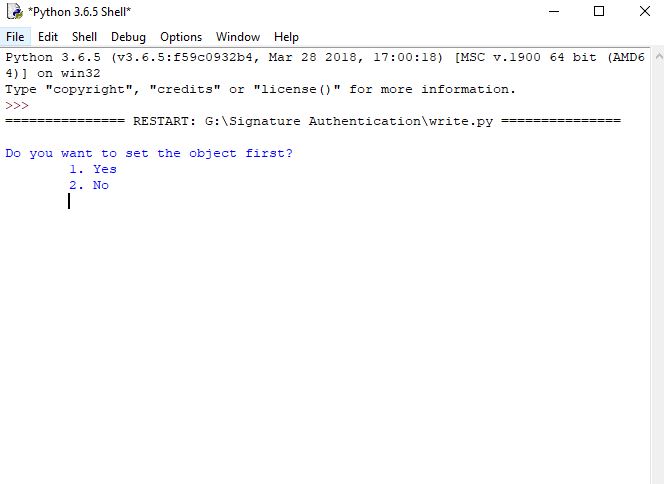
break

cap.release()

cv2.destroyAllWindows()

else:

old\_values = pickle.load(open("old\_values.p", "rb"))



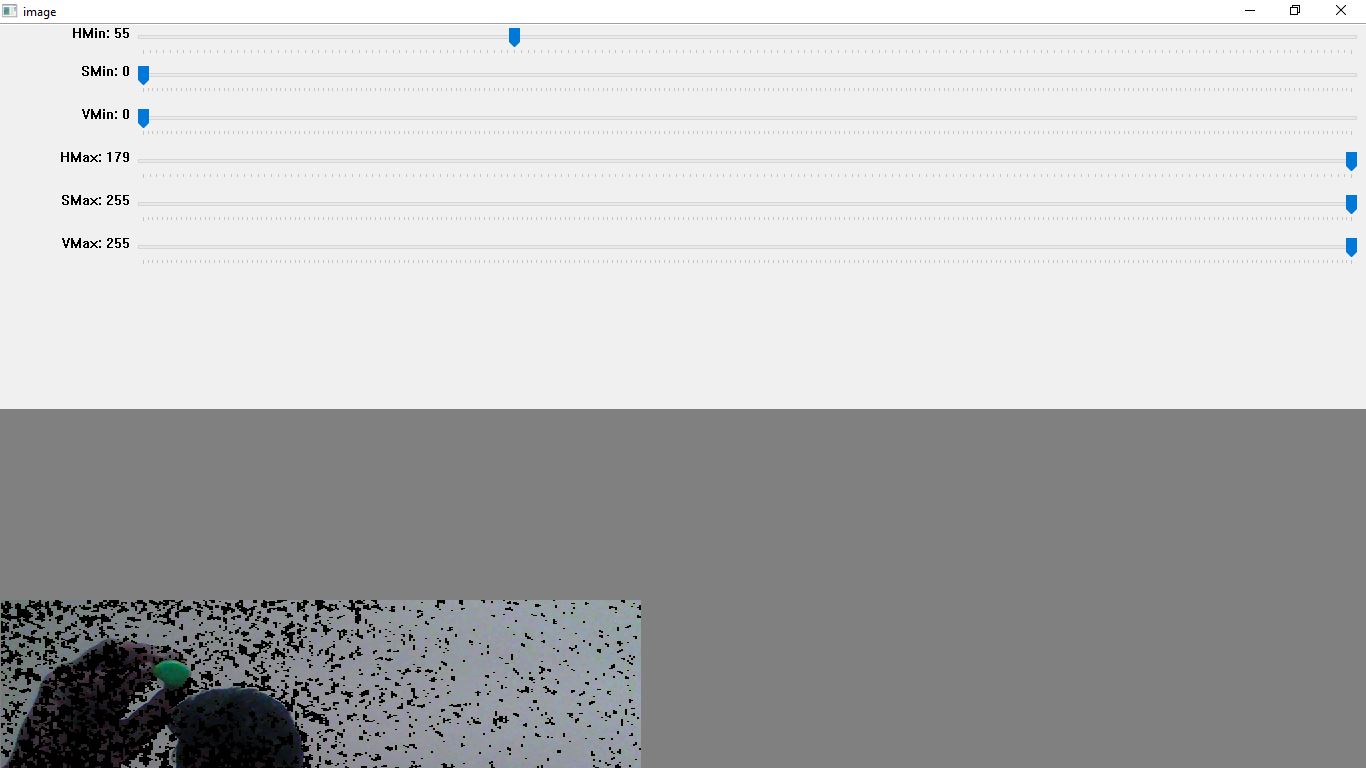


Fig. 8

**Sprint Backlog-2**

**Code for Open Web Cam and Make Dataset**  
 n=1

while(n==1):

cap = cv2.VideoCapture(0)

center\_points = deque()

while True:

\_, frame = cap.read()

frame = cv2.flip(frame, 1)

blur\_frame = cv2.GaussianBlur(frame, (7, 7), 0)

hsv = cv2.cvtColor(blur\_frame, cv2.COLOR\_BGR2HSV)

lower\_blue = np.array([old\_values.get('hMin'), old\_values.get('sMin'), old\_values.get('vMin')])

upper\_blue = np.array([old\_values.get('hMax'), old\_values.get('sMax'), old\_values.get('vMax')])

mask = cv2.inRange(hsv, lower\_blue, upper\_blue)

kernel = cv2.getStructuringElement(cv2.MORPH\_ELLIPSE, (15, 15))

mask = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)

contours, hierarchy = cv2.findContours(mask.copy(), cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)[-2:]

iflen(contours) > 0:

biggest\_contour = max(contours, key=cv2.contourArea)

moments = cv2.moments(biggest\_contour)

centre\_of\_contour = (int(moments['m10'] / moments['m00']), int(moments['m01'] / moments['m00']))

cv2.circle(frame, centre\_of\_contour, 5, (0, 0, 255), -1)

ellipse = cv2.fitEllipse(biggest\_contour)

cv2.ellipse(frame, ellipse, (0, 255, 255), 2)

enter\_points.appendleft(centre\_of\_contour)

fori in range(1, len(center\_points)):

b = random.randint(230, 255)

g = random.randint(100, 255)

r = random.randint(100, 255)

ifmath.sqrt(((center\_points[i - 1][0] - center\_points[i][0]) \*\* 2) + (

(center\_points[i - 1][1] - center\_points[i][1]) \*\* 2)) <= 50:

cv2.line(frame, center\_points[i - 1], center\_points[i], (b, g, r), 4)

cv2.line(mask, center\_points[i - 1], center\_points[i], (b, g, r), 4)

cv2.imshow('Original', frame)

cv2.imshow('Mask', mask)

filename = "outputs/sign\_%d.jpg"%ocount

cv2.imwrite(filename, mask)

k = cv2.waitKey(5) & 0xFF

if k == 27:

break

if k == 113:

ocount+=1

pickle.dump(ocount, open("ocount.p", "wb"))

n = 2

break

cv2.destroyAllWindows()

cap.release()

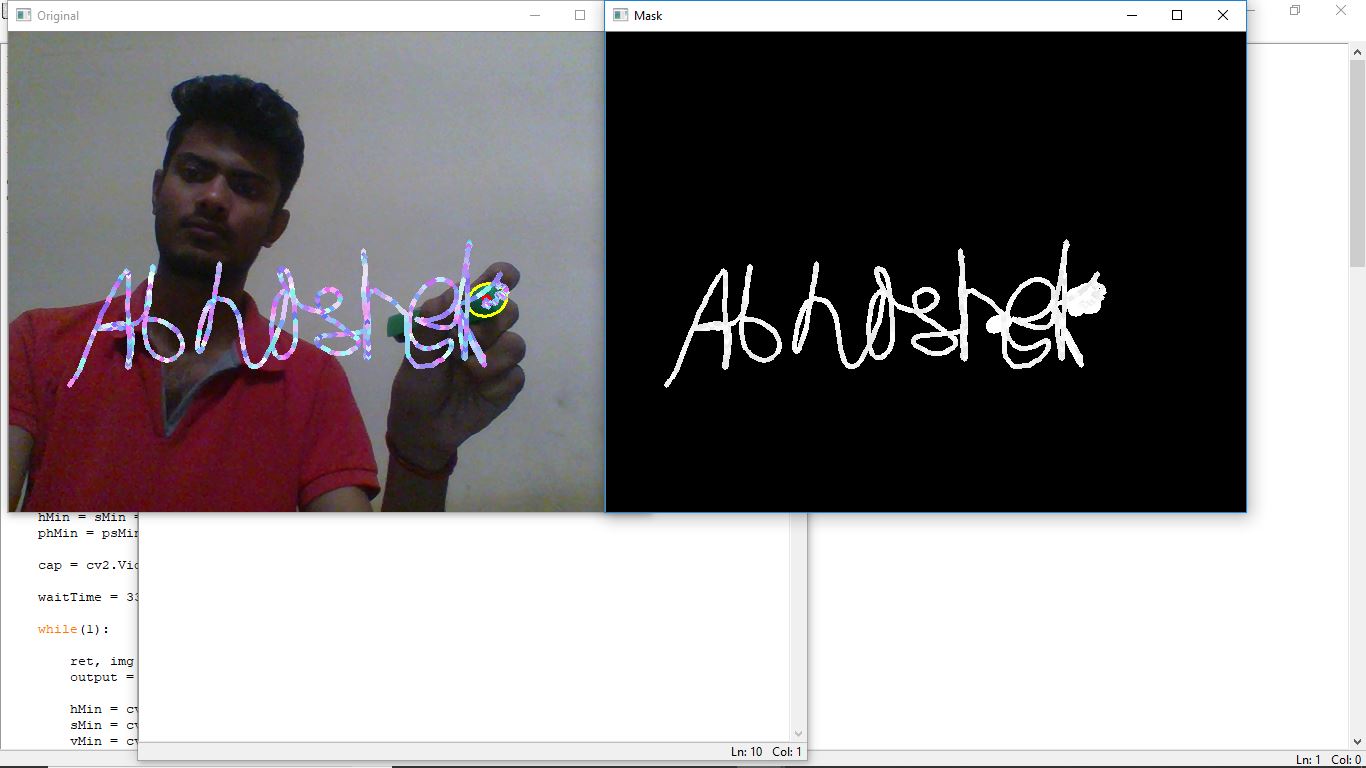


Fig.9

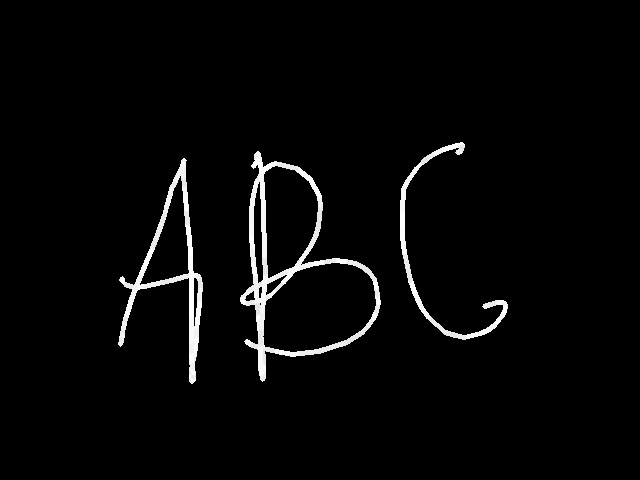
**Sprint Backlog-3**



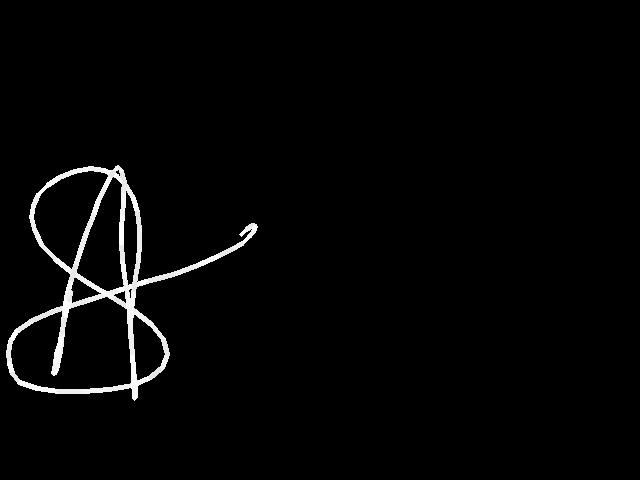
Fig.10



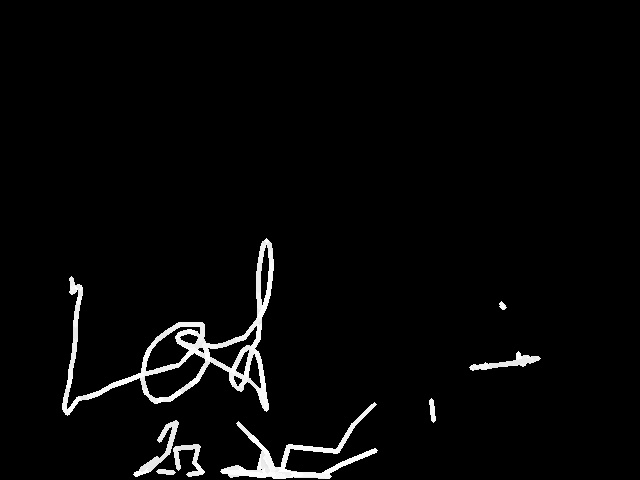












**Sprint Backlog-4**

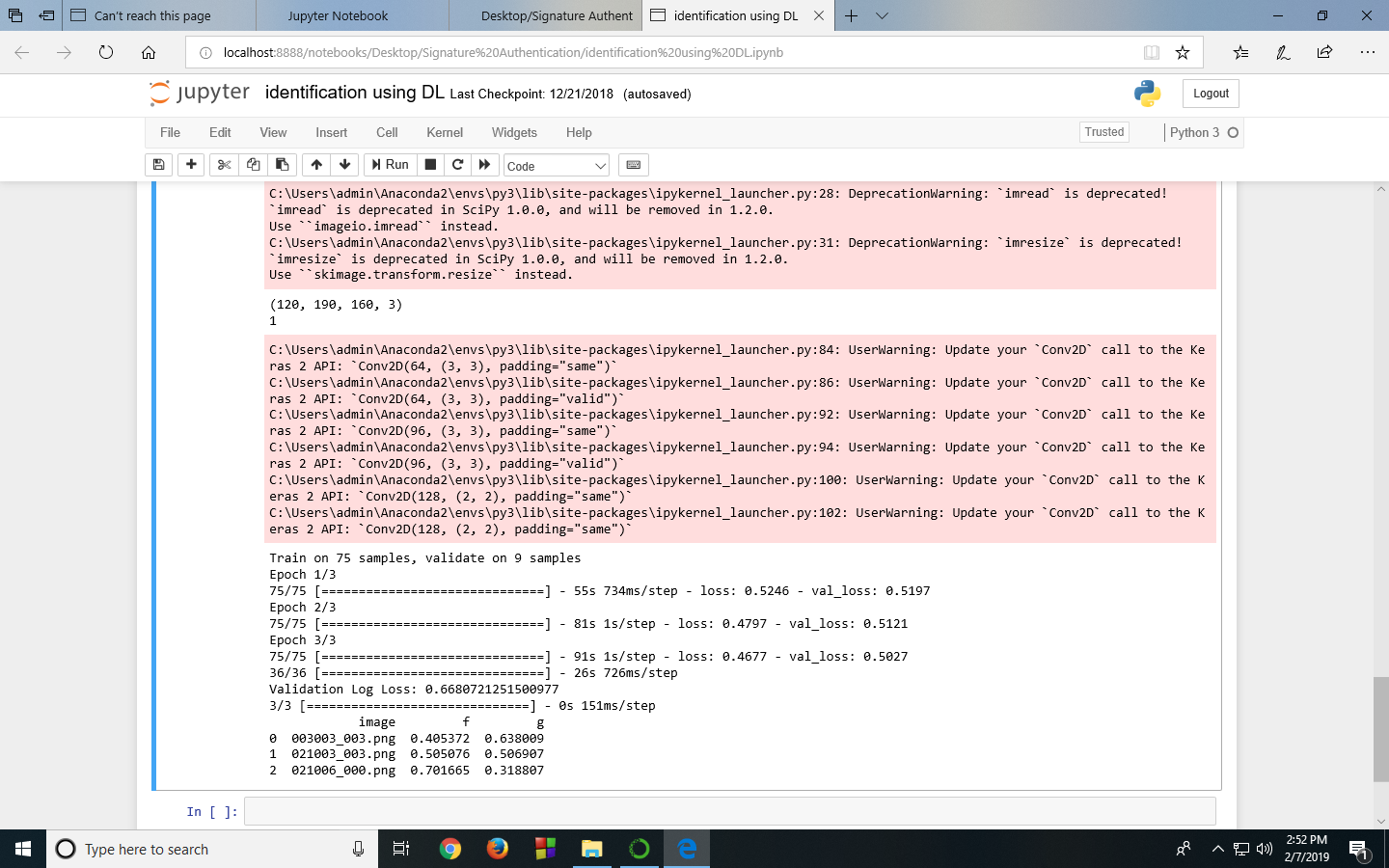


Fig 11

# CHAPTER 5

# CONCLUSION

## Results

We have successfully developed the application for signature authentication. This application provides a hassle free authentication without physical interaction with the device. There are various stages through which it is developed and last a complete project including graphical user interface is developed.

This project is developed to identify the forged signature.

* We developed a deep learning algorithm using Convolutional Neural Network (CNN).
* On the standard dataset we got an accuracy of 71 %.
* While training the model on our own dataset we got an accuracy of 63%.
* The algorithm distinguish the forged and genuine signatures and verifies the person.

## Conclusion

## In this era the authentication of persons is most important because every day we see the news regarding the data breaches. So for that signature authentication process is essential with other authentication process such as face and finger scanning.

## So this module or application will use with the other authentication processes to get the better results and avoid breaches.

## Future Scope

Signature Authentication can be used-

* Defence
* Airport Authority
* Banking and Finances
* Accuracy can be further improved to a good number by increasing the number of layers.
* It can be done on computer having a very good processer.
* So this module or application can be use with other authentication processes to get the better results and avoid forgery.

# 

# ANNEXURES

## www.github.com

## url-<https://github.com/ankitkumawat98/HEX>

# Repository name: -[ankitkumawat98](https://github.com/ankitkumawat98)/[HEX](https://github.com/ankitkumawat98/HEX)

project name:-Signature Authentication in Air

This repository contains all the commits for the projects. It contains the product backlog and sprint backlog and burndown chart.

## References

## Research Paper-1

**A Review of Signature Detection and Tracking from Video Sequences**

Ankit Kumawat1, Lokesh Choudhary2

1Student B.Tech., Department of Computer Engineering, Poornima Institute of

Engineering & Technology, Jaipur

2Student B.Tech., Department of Computer Engineering, Poornima Institute of

Engineering & Technology, Jaipur

[*2015pietcsankit017@poornima.org*](mailto:2015pietcsankit017@poornima.org)

[*2015pietcslokesh@poornima.org*](mailto:2015pietcslokesh@poornima.org)

# *Abstract*

In today’s era real time moving object detection and tracking is one of the important research fields. The steps are used for object recognition are object detection its tracking and identification. In this paper we tracked a signature in air in front of webcam as object is tracked in video sequences. Object tracking is important for computer vision applications. The detection of moving signature in video streams is the first step of information and after that background subtraction is a very popular approach for foreground segmentation.

Tracking of objects is required for security, safety and site management. Object tracking is a Technology that automatically processes those images to detect problematic situations or unusual behavior of human or object. There is a variety of object tracking, detection and identification algorithms which can be used to tracked the Signature. This paper proposes a systematic review of these algorithms and experimental of signature detection and tracking.

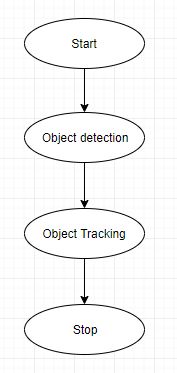
**Keywords:** Signature recognition, Signature detection, Signature identification, Signature tracking.

**1. Introduction**

This paper proposes research conducted so far for Signature detection, tracking and identification of Signature in video streams. Security system is a process of monitoring the behavior and changing information about human’s Signatures. Tracking of signature is the process of determining the characters of interest within a sequence of frames, from its first appearance to its last one. For tracking of objects by detection, the major challenge is how to associate noisy object detections in the current video frame with previously tracked objects.

The steps to be followed to track the objects are as follows.

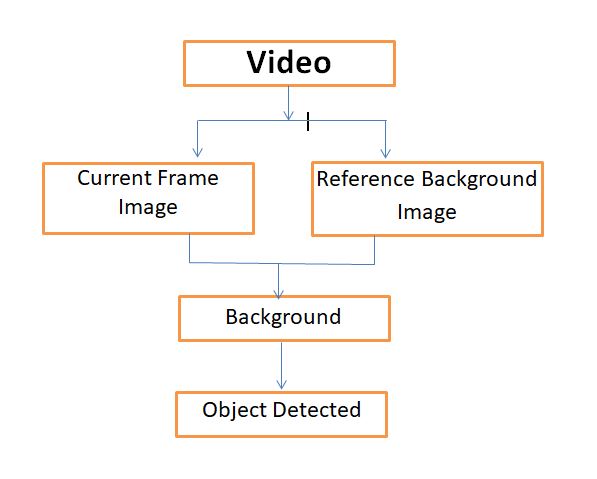
1. Object Detection
2. Object Tracking



**Fig 1: Flow of Procedure**

1. **Object Detection**

Object detection is the process of finding instances of real-world objects by their shape and features such as outline, boundaries, geometrical shapes and points. Object detection algorithms typically use extracted features and learning algorithms to recognize instances or shape of an object and categorized it according to their shape. The major applications of object tracking are image retrieval, security, surveillance, and advanced driver assistance systems.



**Fig 2: Flow chart of Moving Object Detection**

1. **Object Tracking**

Object tracking is the process of assigning an unique identity of an object in the object detection phase and after that tracking the object with the unique identity. The tracking process is to track the assign id in next image frames or video.

Through this unique id we can distinguish the objects and count the objects in a video.

There are many different algorithms which are used for tracking objects. Some of them handle single object tracking while other handle the case of occlusion to make it possible to track multiple objects.

* 1. Point Tracking
  2. Kernel Tracking
  3. Silhouette Tracking

**Table 2:** Comparison of object tracking Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tracking Methods** | | **Algorithms** | **Computational Time** | **Accuracy** |
| Point Tracking | Kalmal filter | Kalmal filtering | Low | Moderate |
| Particle  Filter | Recursive Bayes filtering | moderate | High |
| Multiple Hypothesis | MHT algorithm | low | Low |
| Kernel Tracking | Simple template matching | Matching region of interest | low | Moderate |
| Mean shift method | Optimal gradient decline | Low | Moderate |
| Support vector machine | Positive & negative training values | Moderate | Moderate |
| Layering based tracking | Shape representation using intensity | Moderate | High |
| Silhouette Tracking | Contour Matching | Gradient Descent Algorithm | Moderate | High |
| Shape Matching | Hough  Transform | High | High |

**2. Literature Review**

Recognition of an object from an image frames or video sequences has been one of the most researched fields. Many Algorithms have been introduced but few of them have acceptable because of their high accuracy. Further the real world implementation of these algorithms is bounded by constraints. This paper purposed an innovative technique for object tracking. The properties of images in area like brightness, luminance. Object properties can vary with time and containing many objects of different type, shape and size. This paper deals with the steps involved in object detection and object tracking.

Isaac Cohen & Gerard Medioni [1], In this paper they address a Problem of object detection and object tracking and provide a solution of object tracking in a video stream obtained from a moving platform. This method uses a graph representation. And using this representation the objects are characterized. They demonstrate the results on different real sequences. Then they quantify the results and it shows how tracking overcome detection errors related to object.

Yu Zhong, Anil Kumar Jain, [2], In this Paper A prototype based deformable template models is use for object tracking. To track an object in an image sequence, they use criteria mainly: frame to frame deviation of the object shape and fidelity of the modeled shape to the input image. The image information are used to track the object. Edge, Region consistency and Interframe motion. The tracking proceeds by optimizing an objective function.

Kartik Harihrakrishnan and Dan Schonfeld [3], In this Paper they proposed a fast tracking algorithm that finds an object in first frame and then predicts the object contour. The object contour is predicted by using motion vector information. Tracking is achieved by predicting the object geometries. An adaptive block based approach is used for estimating motions between frames of images.

|  |  |
| --- | --- |
| **Author name** | **Solution proposed** |
| Isaac Cohen & Gerard Medioni | Object tracked in a moving airboerne plateform and graph representation |
| Yu Zhong, Anil Kumar Jain | Criterian based approach uses frame to frame deviation of the object shape and fidelity of the modeled shape |
| Kartik Harihrakrishnan and Dan Schonfeld | It uses a Fast tracking algorithm |

**3. Experimental**

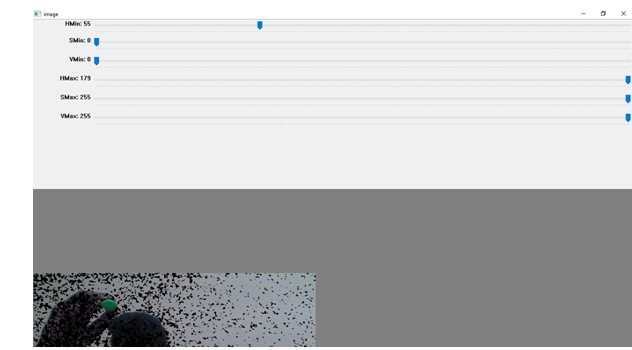
This paper is presented after the experimental of signature tracking using the object tracking methods we have used the silhouette tracking methods in which the contour matching algorithm is used. Contour matching algorithms is uses the Gradient Descent Algorithm. Gradient descent algorithm used for finding the minimum of a function.

Steps to be used in character tracking:

1. Open the webcam
2. Set color constraints
3. Write in front of the webcam

To set the color constraints the HSV values are change to check the better visibility of the object. We need an object that is to be used to write in front of the webcam. And the color constraints are changes according to the object color. HSV stands for ‘hue saturation values’. The min and max Values of color are set for object.

**Libraries used:**

1. Numpy
2. Pandas
3. Matplotlib
4. Train-test split
5. OpenCV
6. Sklearn
7. KERAS
8. 

**Fig 6:** Contour matching

Silhouette Tracking are divided in to two types

* Shape Matching
* Contour Matching
* **Shape Matching:** Shape matching can be performed tracking where an object silhouette. It searches in the current frame and track in next frames which is similar to tracking based on template matching. The search is performed by computing the similarity of the object with the model.
* **Contour Matching**: Contour defines the object boundaries. contour matching is used by applications like image database categorization.

1. **Conclusion**

In this paper the process of signature tracking based on object tracking and techniques are used. After that the comparative study of various techniques used in signature tracking. There are many approaches used to increase the object detection with fewer errors. Furthermore, tracking the signature from the video frames using object tracking algorithms and Experimental of signature tracking using silhouette matching methods and types of silhouette tracking methods. We have identified the limitations and future scope of the signature tracking methods.

**5. References**

**[1]** Review Paper on Moving Object Tracking in Video Processing Jaydeep S. Dadhaniya1 Prof. Kirit D.,2Department of Electronics & Communication Engineering 1,2Nobel Group of Institution-Junagadh, Gujarat, India-2015.International Journal for Scientific Research & Development 01, 2015

**[2]** A Review of Detection and Tracking of Object from Image and Video Sequences, Mukesh Tiwari, Department of Electronics & Communication, Rajiv Gandhi Technical University, Bhopal, International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 5 (2017)

**[3]** A Comparative Study of Object Tracking Techniques Meha J. Patel1, Bhumika Bhatt2, International Journal of Innovative Research in Science, Engineering and Technology 3, March 2015.

## Research Paper-2

**Signature Verification by Neural Network Approach**

**Abhishek Sharma**

Department of Computer Engineering, Poornima Institute of

Engineering & Technology, Jaipur

[*2015pietcsabhishek006@poornima.org*](mailto:2015pietcsabhishek006@poornima.org)

# *Abstract*

Signature verification is a technique used by banks, government agencies and high-profile institutions to validate the identity of an individual. Signature verification is often used to compare signatures in banks and offices. Signature verification plays a vital role in the field of the commercial, financial and for the legal matters. Any person’s signature is considered as his approval for any work that’s why the signature is the preferred authentication method. Signature verification is done in this paper by techniques of image processing, geometric feature extraction and by using neural network technique.

**Keywords:**Signature verification, Authentication, Matching Technique.

# Introduction

For validation of any legal document, signature is used. So the need of the signature increases. The handwritten signatures are unique and it’s almost impossible to copy and do forged signature. The main advantage of signature verification has over other signature verification methods is that everyone already commonly accepts it. This signature verification system extracts features from scanned signature image. The features extracted in this methodology are independent. In this only pixel image needs to be evaluated. But it’s difficult to develop signature verification system based on image as we don’t have as many as desirable features like order of strokes, velocity, pressure etc. The verification process fully relies on the static signature images. In the area of handwritten signature verification, different technologies have been introduced and still the area is evolving.

# Preprocessing

We solve the problem in two steps. First of all the scanned signature image is go through the preprocessing so that a computer can process. Then the features extracted from it that can distinguish forged signatures from the genuine signature. Data processing performed on raw data to prepare and make the data ready for further processing. Hence, the preprocessing is an initial step that transforms the data into data that will be less complex and easily processed.

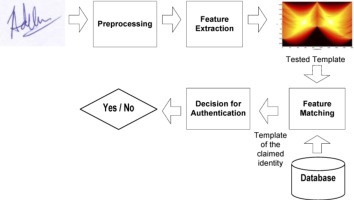


Fig 2.1: Steps in Signature Verification

An image of signature is captured first and converted into a format that can be processed by a computer. In first step is to convert the RGB image to into a gray scale image. It helps in reducing the complexity and run time. System does less processing on a gray scale image than a RGB image. Then using a cropping algorithm image is cropped to get precisely only the area of signature, which is the actual region of interest from the complete image.

The data we are using to train the network contains images like following:



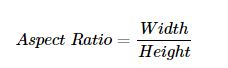
Fig 2.2 : Sample data images

# Feature Extraction

Feature extraction is the main key to develop signature verification system. We use a set of 4 global features. The features are independent that cannot be affected by temporal shift. These are shape and dimensional based features of a signature image. Following are the features:

* Convex Hull area / Bounding Rectangle area
* Aspect ratio of bounding rectangle
* Contour area
* Perceptual hash of the image

1. **Convex Hull area / Bounding Rectangle area:** Convex hull will look similar to the contour approximation. It will show the same results as the contour approximation shows. Cv.convexHull() Function is used in python to check a curve for convexity defect and corrects it.
2. **Aspect ratio of bounding rectangle:** It is the ratio of width to height of bounding rectangle of the object. It is calculated by

****

1. **Contour area:** cv.drawContours function is used for drawing the contours. It can also be used to draw any shape which is having its boundary points. Its first argument is source image, second argument is the contours.
2. **Perceptual hash of the image:** Image hashing or perceptual hashing is the process of examining the contents of an image and constructing a hash value that uniquely identifies an input image based on the contents of an image.

****

Fig 3**:** hash function working

# Neural Network Approach

Pattern recognition is the power of Neural Networks and this is one of the reasons behind widespread usage of neural networks. Artificial Neural Networks are mathematical models inspired by biological neurons. It is a network of interconnected artificial neurons with a complex relationship to find pattern in the data. The neural network approach first extracts the features described above from the signature images from various signers. Then it finds and learns the relationship between the signatures in dataset and class into forged or genuine signature. After finding the relation, the network tests the signature and classifies it into the forged or genuine according to the person. That is why it is the best approach for signature verification.

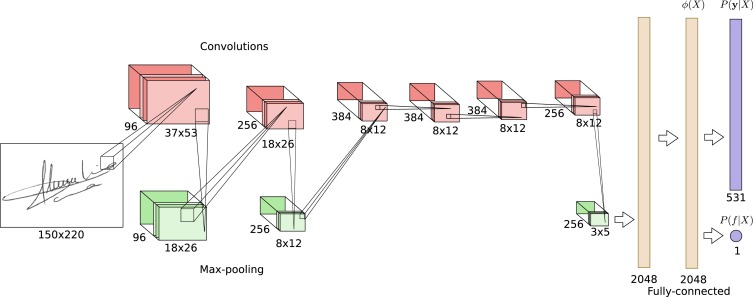


Fig 4.1: Convolutional Neural Network

We used Convolutional Neural Networks (CNNs) here in this approach. CNNs are proven to be successful and used vastly in deep learning tasks. In other methods of signature verification, some features extracted from signature image and fed into a classifier to arrive at a classification decision.

# Conclusion:

This paper includes the experimental of Signature Verification systemWe experimented with several variations on signature verification tasks. We showed that convolutional neural networks do an excellent job of verifying signaturesand for training and testing of the system different genuine and forged signature are used so that the output may be good.The results provided in this research used a total of 100 signatures. These signatures are comprised of 100 different people and for each person there are 3 samples of genuine signature and 3 forged signature. The features extracted from 3 genuine and 3 forged signatures for each person were used to train a neural network. The accuracy of the system is 83.2%.

# Reference:

**[1]** Signature Recognition and Verification with Artificial Neural Network Using Moment Invariant Method Cemil Oz Sakarya University Computer Engineering Department, Sakarya, Turkey [ozc@umr.edu](mailto:ozc@umr.edu).

**[2]**Offline Handwritten Signature Verification using Neural Network Pallavi V. Hatkar Department of Electronics Engineering, TKIET Warana, India Prof.B.T.Salokhe Department of Electronics Engineering, TKIET Warana, India Ashish A.Malgave Department of Electrical Engineering, AMGOI Wathar, IndiaJAN-2015.

**[3]**Offline Signature Verification with Convolutional Neural Networks Gabe Alvarez galvare2@stanford.edu, Blue Sheffer bsheffer@stanford.edu, Morgan Bryant [mrbryant@stanford.edu](mailto:mrbryant@stanford.edu)

**[4]**Offline signature recognition using neural networks approach Ali Karounia \*, Bassam Dayab , SamiaBahlakb a Lebanese University, EcoleDoctorale des Sciences et de Technologie (EDST), LEBANON b Lebanese University, Institute of Technology of Saida, LEBANON, P.O.B. 813

**[5]**SIGNATURE VERIFICATION SYSTEM ,W. Robert Carnes, 32 Phillips La., Darien, Conn, 0.6820,Feb. 21, 1984

**[6]**A review of Handwritten Signature Verification Systems and Methodologies, G. Padmajadevi Associate professor,E&CDeptMalnad College Of Engineering Hassan, India padmajadevig@gmail.com, Dr.K.S. Aprameya Professor, E&E Dept UBDT College Of Engineering Davangere, India aprameya\_ks@yahoo.com-International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016

**[7]**An Evaluation on Offline Signature Verification using Artificial Neural Network Approach, Othman o-khalifa, Md. KhorshedAlam Department of Electrical and Computer Engineering International Islamic University Malaysia, Kuala Lumpur, Malaysia khalifa@iium.edu.my, shishir\_lmu@yahoo.com, Aisha Hassan Abdalla Department of Electrical and Computer Engineering International Islamic University Malaysia, Kuala Lumpur, Malaysia aisha@iium.edu.my- 2013 INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE)

## DST Document.

# Title of the Project : Signature Verification in Air

* **Project Summary:**

To solve this problem we are developing a software based on Artificial Intelligence which will automatically check whether the signature is of the same person or not and it will be more accurate. Another feature is that you can write in the air in front of webcam without physically touching the device.

* **Technical Details of the Projects :**
* Origin of the Proposal :

Today signature Verificationis necessary in each and evey field and doing it manually always carries chances of error and false verification of just 1 of the 100 cases can cost crores sometimes. It also consumes too much time and waste human resources.

* Definition of the problem :
* Risk in manual process
* Too much time consuming
* Wastage of human resources
* Problem in device interface

Objectives :

To provide an AI based software solution which can automate the process and check accurately that whether the signature is of right person. So that we can reduce the time in process and free the resources. Also a person can easily write on the screen without physicallhy interacting to the screen or device. It means you can wirte in the air in front of webcam.

Work Plan :

1. Stage 1: An idea for a project will be carefully examined to determine whether or not it benefits the organization. During this phase, a decision making team will identify if the project can realistically be completed.
2. Stage 2: Study the technology that can be used to measure it and the requirements for this technology.
3. Stage 3: Learn the various tools for designing the prototype of the software.
4. Stage 4: Learn the programming languages and database and analyse it for the working condition of Software.
5. Stage 5: Development of the software and train it.
6. Stage 6: Final testing and put out in the real world.
7. Stage 7: Report writing.

* Methodology:

Research work for collecting information about the technology and requirements involved in the project.

Deep learning model will be identified in this phase.

First module of signature without physical interaction will be developed using Python.

Choosing the model and implement it according requirement.

Training of the deep learning model over the dataset will be done in this phase.

Testing of the software will be done in this phase.

Deployment of the final software in the market.

* Organisation of work element :

Research and literature viewing 🡪 Requirements of technology being used🡪 Selection of Deep learning model 🡪 Writing programming code in Python programming language 🡪 Training of the model over the dataset 🡪 Implementation of software 🡪 Initial testing 🡪 Review(Correct mistakes if any found) 🡪 Final testing 🡪 Deploying it in real world.

* Time schedule:
  + - 1. Literarture viewing : ½ Month
      2. Model Selection : ½ Months
      3. Software building : 1 Months
      4. Collection and Training: ½ Months
      5. Testing & Deployment : 1 Months
* Proposed outcome/findings:
* More accurate checking and verifying
* Outcome as an AI software
* Signature varification
* Easy to interact with device
* Improve the productivity
* Reduced human resources
* **Details of facilities to be provided by the Institution:**
* Computing facility provided.
* Research laboratories.
* General software required.
* Teacher’s guidance.
* **Budget Estimates :**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | COMPONENT NAME | QUANTITY | COST |
| 1 | Domain | 1 | 2000 |
| 2 | Internet | - | 2000 |
| 3 | HTTPS Server Security | 1 | 700 |
| 4 | Lan base system | 1 | 2000 |
| 5 | AWS for Server | 1 | 6960 |
| 6 | Report Writing | 5 | 1050 |
| 7 | Bulk Storage | 1 | 1620 |

* **Utilization of the outcome of project :**
* More accurate system
* Easy interaction with device
* Automation of whole system of signature authentication
* Not require human resources
* It can save banks from losing crore of money.