SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Pattern Detection and Recognition using Deep Learning

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Faizan Shaikh Abdul Khalil Shaikh	Exam no: B120054285
Manasi Sajan Khapke	Exam no: B120054352
Shraddha Balasaheb Gunjal	Exam no: B120054309

Under The Guidance of

Prof. D.T. Mane

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Pune Institute of Computer Technology, Sr. No 27, Pune-Satara Road, Dhankawadi, Pune, Maharashtra 411043

Pune Institute of Computer Technology
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Pattern Detection and Recognition using Deep Learning

Submitted by

Faizan Shaikh Abdul Khalil Shaikh
Manasi Sajan Khapke
Shraddha Balasaheb Gunjal

is a bonafide work carried out by Students under the supervision of Prof. D.T.Mane and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering) Project.

Prof. D.T.Mane Internal Guide Dept. of Computer Engg.	Prof. G. P. Potdar H.O.D Dept. of Computer Engg.

Abstract

Pattern detection and recognition is one of the interesting and challenging research topics. Here, we tackle a problem to automate the process of recognizing endangered right whales from aerial photographs. The current software takes a lot of time and efforts as it relies on human intervention. Automating the identification of unique right whale would allow researchers to better focus on their conservation efforts. Our approach for this problem is to use Deep Learning to minimize human labour and achieve better results.

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'Pattern Detection and Recognition using Deep Learning'****.*

I would like to take this opportunity to thank my internal guide ***Prof. D.T. Mane*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Prof. G. P. Potdar****, Head of Computer Engineering Department, PICT for his indispensable support, suggestions.*

Faizan Shaikh Abdul Khalil Shaikh
Manasi Sajan Khapke
Shraddha Balasaheb Gunjal
(B.E. Computer Engg.)

INDEX

1 Synopsis **1**

1.1 Project Title 2

1.2 Project Option 2

1.3 Internal Guide 2

1.4 Sponsorship and External Guide 2

1.5 Technical Keywords (As per ACM Keywords) 2

1.6 Problem Statement 2

1.7 Abstract 3

1.8 Goals and Objectives 3

1.9 Relevant mathematics associated with the Project 3

1.10 Names of Conferences / Journals where papers can be published . . 4

1.11 Review of Conference/Journal Papers supporting Project idea 5

1.12 Plan of Project Execution 8

2 Technical Keywords **9**

2.1 Area of Project 10

2.2 Technical Keywords 10

3 Introduction **11**

3.1 Project Idea 12

3.2 Motivation of the Project 13

3.3 Literature Survey 13

4 Problem Definition and scope **24**

4.1 Problem Statement 25

4.1.1 Goals and objectives 25
4.1.2 Statement of scope 25
4.2 Software context 25
4.3 Major Constraints 26
4.4 Methodologies of Problem solving and efficiency issues 26
4.5 Scenario in which multi-core, Embedded and Distributed Computing used 26
4.6 Outcome 26
4.7 Applications 27
4.8 Hardware Resources Required 27
4.9 Software Resources Required 27

5 Project Plan **28**
5.1 Project Estimates 29
5.1.1 Reconciled Estimates 29
5.1.2 Project Resources 30
5.2 Risk Management w.r.t. NP Hard analysis 30
5.2.1 Risk Identification 30
5.2.2 Risk Analysis 31
5.2.3 Overview of Risk Mitigation, Monitoring, Management . . 31
5.3 Project Schedule 34
5.3.1 Task network 35
5.3.2 Timeline Chart 35
5.4 Team Organization 35
5.4.1 Team structure 35
5.4.2 Management reporting and communication 35

6 Software requirement specification **37**
6.1 Introduction 38
6.1.1 Purpose and Scope of Document 38
6.1.2 Overview of responsibilities of Developer 38
6.2 Usage Scenario 39

6.2.1 User profiles . . . 39
6.2.2 Use-cases . . . 39
6.2.3 Use Case View . . . 40
6.3 Data Model and Description . . . 40
6.3.1 Data Description . . . 40
6.4 Functional Model and Description . . . 41
6.4.1 Data Flow Diagram . . . 41
6.4.2 Description of functions . . . 42
6.4.3 Activity Diagram: . . . 42
6.4.4 Non Functional Requirements: . . . 43
6.4.5 State Diagram: . . . 44

7 Detailed Design Document using Appendix A and B 45
7.1 Introduction . . . 46
7.2 Architectural Design . . . 46
7.3 Data design (using Appendices A and B) . . . 46
7.3.1 Internal software data structure . . . 46
7.3.2 Global data structure . . . 46
7.3.3 Temporary data structure . . . 47
7.3.4 Database description . . . 47
7.4 Component Design . . . 48
7.4.1 Class Diagram . . . 48
7.4.2 Sequence Diagram . . . 48

8 Project Implementation 51
8.1 Introduction . . . 52
8.2 Tools and Technologies Used . . . 52
8.3 Algorithm Details . . . 53
8.3.1 Convolutional Neural Network . . . 53
8.4 Verification and Validation for Acceptance . . . 55

9 Software Testing 56
9.1 Introduction . . . 57

9.1.1 Goals and objectives 57
9.1.2 Statement of scope 57
9.2 Type of Testing Used 57
9.2.1 Unit testing 57
9.2.2 Integration testing 58
9.2.3 Validation testing 58
9.2.4 GUI Testing 58
9.2.5 High-order testing 58
9.3 Test Cases and Test Results 59
9.3.1 Unit Testing 59
9.3.2 Integration Testing 59
9.3.3 Validation Testing 59
9.3.4 High-order Testing 60

10 Results **61**
10.1 Screen shots 62
10.2 Outputs 62
10.2.1 Outputs / Snap shots of the results 62

11 Deployment and Maintenance **66**
11.1 Installation and un-installation 67
11.2 User help 67

12 Conclusion and future scope **68**

Annexure A References **70**

Annexure B Laboratory assignments on Project Analysis of Algorithmic Design **72**

Annexure C Laboratory assignments on Project Quality and Reliability Testing of Project Design **74**

Annexure D Project Planner **76**

List of Figures

5.1 Waterfall model 29

6.1 Use case diagram 40
6.2 Dataset Distribution 41
6.3 Data Flow diagram 42
6.4 Activity diagram 43
6.5 State transition diagram 44

7.1 Architecture diagram 47
7.2 Class Diagram 49
7.3 Sequence Diagram 50

8.1 Basic CNN Architecture D 54

10.1 Input Whale Image 62
10.2 Localizer Output 63
10.3 Classifier Output 63
10.4 Output 1 64
10.5 Output 2 64
10.6 Output 3 65

D.1 Gantt Chart 77
D.2 Gantt Chart 77
D.3 Gantt Chart 78
D.4 Gantt Chart 78

List of Tables

4.1 Hardware Requirements 27

5.1 Hardware Requirements 30
5.2 Risk Table 32
5.3 Risk Probability definitions 32
5.4 Risk Impact definitions 32
5.5 Team Structure 36

6.1 Use Cases 39

9.1 Unit Testing 59
9.2 Integration Testing 59
9.3 Validation Testing 59

B.1 IDEA Matrix 73

CHAPTER 1

SYNOPSIS

1.1 PROJECT TITLE

Pattern Detection and Recognition Using Deep Learning

1.2 PROJECT OPTION

Internal Project

1.3 INTERNAL GUIDE

Prof. D.T.Mane

1.4 SPONSORSHIP AND EXTERNAL GUIDE

N.A.

1.5 TECHNICAL KEYWORDS (AS PER ACM KEYWORDS)

1. C. Machine Learning
    (a) C.2 Deep Learning
        i. C.2.4 Convolutional Neural Network
            A. Sub-sampling
            B. Max Pooling
            C. Feature Extraction
            D. Neural Network
            E. Pattern Detection

1.6 PROBLEM STATEMENT

To automate the right whale recognition process from aerial photographs of individual whales using Deep Learning.

1.7 ABSTRACT

- Pattern detection and recognition is one of the interesting and challenging research topics. Here, we tackle a problem to automate the process of recognizing endangered right whales from aerial photographs. The current software takes a lot of time and efforts as it relies on human intervention. Automating the identification of unique right whale would allow researchers to better focus on their conservation efforts. Our approach for this problem is to use Deep Learning to minimize human labour and achieve better results.

1.8 GOALS AND OBJECTIVES

1. To perform analysis of different deep learning techniques for pattern detection and recognition.
2. To implement deep learning algorithm on image dataset for obtaining state-of-the-art performance.
3. To obtain the optimized result by increasing efficiency of deep learning algorithm.

1.9 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT

System Description:

- Start: Dataset of Images
- End: Detection and Recognition of Pattern
- Input: X = (In where I is image and n belongs to real numbers)
- Output: Y = (Ci where C is the class of pattern)

- Functions : F = (f1, f2, f3, f4, f5)
  - f1: function for data preprocessing
  - f2: function for defining neural network architecture
  - f3: function for feature extraction
  - f4: function for pattern detection and recognition
  - f5: function for image classification

- Success Conditions: Image successfully classified according to category

- Failure Conditions: Error in classification

- Deterministic Data:
  1. Label of Image
  2. Image Resolution

- Non-Deterministic Data:
  1. Pattern Orientation
  2. Viewpoint of Image

1.10 NAMES OF CONFERENCES / JOURNALS WHERE PAPERS CAN BE PUBLISHED

1. Conference on Computer Vision and Pattern Recognition (CVPR)
2. International Conference on Machine Learning (ICML)
3. International Conference on Advances in Pattern Recognition (ICAPR)
4. International Journal of Computer Applications(IJCA)
5. International Journal of Computer Science and Engineering (IJCSE)

1.11 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT IDEA

Sr. No.	Title	Summary
1.	Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012	This paper introduces the Deep Convolutional Neural Network technique for image classification. It achieved Top-1 and top-5 error rates of 37.5% and 17.0% on ImageNet dataset.
2.	Bai, Jinfeng, et al. "Image character recognition using deep convolutional neural network learned from different languages." Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 2014.	This paper proposed a shared-hidden-layer deep convolutional neural network (SHL-CNN) for image character recognition. In SHL-CNN, the hidden layers are made common across characters from different languages, performing a universal feature extraction process that aims at learning common character traits existed in different languages such as strokes, while the final softmax layer is made language dependent, trained based on characters from the destination language only. Experiments on English and Chinese image character recognition. reduce errors on both the tasks by 16-30% relatively.

3.	Goodfellow, Ian J., et al. ”Multi-digit number recognition from street view imagery using deep convolutional neural networks.” arXiv preprint arXiv:1312.6082 (2013).	Application of DistBelief implementation of deep neural networks in order to train large, distributed neural networks on high quality images. The performance of this approach increases with the depth of the convolutional network. Accuracy was 99.8% on recaptcha.
4.	Kim, In-Jung, and Xiaohui Xie. ”Handwritten Hangul recognition using deep convolutional neural networks.” International Journal on Document Analysis and Recognition (IJDAR) 18.1 (2014): 1-13.	Hangul recognizers are built based on deep convolutional neural networks, and propose several novel techniques to improve the performance and training speed of the networks. Result: 95.96% on SERI95a and 92.92% on PE92, 99.71% on MNIST.
5.	Ciresan, Dan, Ueli Meier, and Jrgen Schmidhuber. ”Multi-column deep neural networks for image classification.” Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.	On many image classification datasets the MCDNN improves the state-of-the-art by 30-80%.This drastically improve recognition rates on MNIST, NIST SD 19, Chinese characters, traffic signs. This method is fully supervised and does not use any additional unlabeled data source. Single DNN already are sufficient to obtain new state-of-the-art results; combining them into MCDNNs yields further dramatic performance boosts.

6.	Gao, Shenghua, et al. ”Single Sample Face Recognition via Learning Deep Supervised Auto-Encoders.”, in IEEE Transaction (2013).	This paper targets learning robust image representation for single training sample per person face recognition. This proposes a supervised autoencoder, which is a new type of building block for deep architectures. It enforce the faces with variants to be mapped with the canonical face of the person, for example, frontal face with neutral expression and normal illumination; also enforce features corresponding to the same person to be similar.
7.	Sermanet, Pierre, Sandhya Chintala, and Yann LeCun. ”Convolutional neural networks applied to house numbers digit classification.” Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.	They augmented the traditional ConvNets architecture by learning multi-stage features and by using Lp pooling and establish a new state-of-the-art of 95.10% accuracy on the SVHN dataset (48% error improvement).
8.	Claudiu Ciresan, Dan, et al. ”Deep big simple neural nets excel on handwritten digit recognition.” arXiv preprint arXiv:1003.0358 (2010).	Techniques used: Back propagation NeuralNetwork, MLP (Multilayer Perceptron). Result: The best network has an error rate of only 0.35% (35 out of 10,000 digits)Which is very efficient as compared to previous error rates.

1.12 PLAN OF PROJECT EXECUTION

- Survey Phase (July - December)
  - Topic discussion (1st week of July - 4th week of August)
  - Literature Survey (2nd week of July - 1st week of December)
  - Problem Statement (2nd week of August 2nd week of September)
- Implementation Phase (September-March)
  - Basic implementation (1st week of September - 4th week of November)
  - Optimization (4th week of September - 4th week of February)
  - Testing (1st week of January - 1st week of March)

CHAPTER 2

TECHNICAL KEYWORDS

2.1 AREA OF PROJECT

Machine Learning

2.2 TECHNICAL KEYWORDS

1. C. Machine Learning
    (a) C.2 Deep Learning
        i. C.2.4 Convolutional Neural Network
            A. Sub-sampling
            B. Max Pooling
            C. Feature Extraction
            D. Neural Network
            E. Pattern Detection

CHAPTER 3

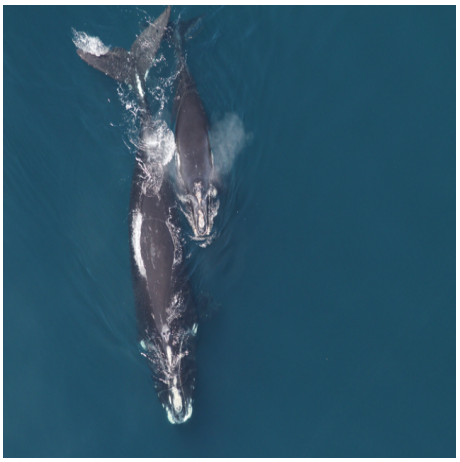
INTRODUCTION

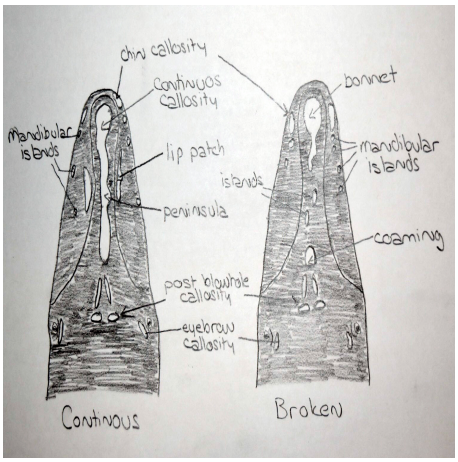
3.1 PROJECT IDEA

- Why do we rely on Machine Learning? To answer this, we should ponder upon the fact that we strive for automation from computers. We teach computers to do human-like actions ranging from building a car in an automobile industry, to driving a rover on Mars. We do this by first defining a mathematical rule which a rational agent has to follow to come up to a solution. Then we train a model built on these rules, with relevant information for the model to function according to our needs. We then apply this trained model for a variety of applications. Thus we can infer that Machine Learning can be a great tool in automation industry.

- To recognize the whale from aerial photographs provide a great challenge to computer programmers. This task cannot be done efficiently because of low resolution, blurring, distortion, luminance variance, complicate background, etc. To overcome these all degradations deep leaning can be used efficiently as deep learning depends on the features extracted from the image at various stages and not on the handcrafted features. Hence, pattern detection and recognition using deep learning serves a very good idea for working.

- Right whale is an endangered species with fewer than 500 left in the Atlantic Ocean. Each right whale has unique callosity pattern on the head.





3.2 MOTIVATION OF THE PROJECT

- Dream of AI is to make computers as intelligent as Humans is the thing which most of the scientists consider as a challenge. The computer should be able to work as efficient as humans in all conditions and give proper results. The machine learning plays an important role in training computers on the different data and using the learned algorithm for further use. The next step of machine learning is the deep architectures which work well for vision, audio, NLP, etc.
- Deep learning gives us the ability to realize patterns in a given scenario and make the right decision about how to proceed. Also it had proved its efficiency in handwritten digit and character recognition and further can be extended to extract pattern from natural scene.
- In todays IT industry, deep learning is the Gold Rush! Due to its ability of handling large amount of data successfully and efficiently, it has become the choice of industry.

3.3 LITERATURE SURVEY

Sr. No.	Year	Title	Conference/ Journal Name	Summary	Technique	Result

1.	2012	ImageNet Classification with Deep Convolutional Neural Networks	NIPS	This paper introduces the Deep Convolutional Neural Network technique for image classification. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, they used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers they employed a recently-developed regularization method called dropout that proved to be very effective.	Deep Convolutional Neural Network	Top-1 and top-5 error rates of 37.5% and 17.0%

2.	2012	Recognizing Handwriting Indian Numbers using Neural Network	International Journal of Scientific Knowledge (Computing and Information Technology)	This paper is concerned with proposing technique for recognition a handwritten Indian numbers by using artificial neural network technique. This technique is based on multilayer feed forward back propagation neural network. The experiments are conducted using two different sets of different sizes of training sets.	Multi-layer feed forward back propagation neural network	The generalization ratio of recognition of digits for testing set was 84.03 by using 1500 samples and 92.31 for testing set by using 2500 samples. As size of the training set is increased, the results of recognition digits using neural network is also improved.

3.	2015	Deep Convolutional Network for Handwritten Chinese Character Recognition	-	This paper explored the performance of deep convolutional neural network on recognizing handwritten Chinese characters. They ran experiments on a 200-class and a 3755-class dataset using convolutional networks with different depth and filter numbers.	Deep Convolutional Neural Network	98.1% accuracy on Top 1 and 99.7% on Top 5

4.	2013	Better digit recognition with a committee of simple Neural Nets	ICDAR IEEE	This paper proposes a new method to train the members of a committee of one-hidden-layer neural nets. Instead of training various nets on subsets of the training data, they preprocess the training data for each individual model such that the corresponding errors are decorrelated.	Committee of Neural Network	On the MNIST digit Recognition benchmark set, the obtained recognition error rate was 0.39%, using a committee of 25 one-hidden-layer neural nets, which is on par with state-of-the-art recognition rates of more complicated systems.

5.	2012	An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition	IEEE Transactions on Industrial Electronics	This paper presents an accelerometer-based digital pen for handwritten digit and gesture trajectory recognition applications. The features reduced from trajectory detection are sent to a trained probabilistic neural network for recognition. The experimental results have successfully validated the effectiveness of the trajectory recognition algorithm for handwritten digit and gesture recognition using the proposed digital pen.	Probabilistic Neural Network	The overall handwritten digit recognition rate was 98%, and the gesture recognition rate was also 98.75%.

6.	2011	DIGIT RECOGNITION SYSTEM USING BACK PROPAGATION NEURAL NETWORK	International Journal of Computer Science and Communication	In the proposed work, back propagation neural network based digit recognition system has been developed. When digit image is scanned, quality of image is degraded and some noise is added into this image. So it is necessary to reduce the noise and improve the quality of the digit image for OCR system.. For this, frequency domain based Gausian filter is used to improve the quality and denoise the digit image. The goal of image segmentation is to separate the clear digit print area from the non-digit area. After segmentation, binary digit image is skeleton to reduce the width of digit into just a single line. Experiments have been performed on standard dataset of digits.	Back propagation neural network	Results of this study are quite promising and 96.6% accuracy has been achieved by the proposed system.

7.	2013	Handwritten Character and Digit Recognition Using Artificial Neural Networks	International Journal of Advanced Research in Computer Science and Software Engineering	In this paper, the multilayer perception artificial neural network to recognize the handwritten digits and characters is used. Character reorganization device is one of such smart devices that acquire partial human intelligence with the ability to capture and recognize various characters and digits. In this MLP network is use the back propagation algorithm to train and test the data.	MLP with Back propagation	Here error rate is 1.089

8.	2012	Multi-column Deep Neural Networks for Image Classification	IDSIA	Traditional methods of computer vision and machine learning cannot match human performance on tasks such as the recognition of handwritten digits or traffic signs. But biologically plausible deep artificial neural network architectures can.Several deep neural columns become experts on inputs preprocessed in different ways; their predictions are averaged. On the very competitive MNIST handwriting benchmark, our method is the first to achieve near-human performance. On a traffic sign recognition benchmark it outperforms humans .It also improve the state-of-the-art on a plethora of common image classification benchmarks.	MCDNN	This is the first time human-competitive results are reported on widely used computer vision benchmarks. On many other image classification datasets the MCDNN improves the state-of-the-art by 30-80%.This drastically improve recognition rates on MNIST, NIST SD 19, Chinese characters, traffic signs.

9.	2009	Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks	NIPS	Offline handwriting recognitionthe transcription of images of handwritten textis an interesting task, in that it combines computer vision with sequence learning. In most systems the two elements are handled separately, with sophisticated preprocessing techniques used to extract the image features and sequential models such as HMMs used to provide the transcriptions. By combining two recent innovations in neural networksmultidimensional recurrent neural networks and connectionist temporal classificationthis paper introduces a globally trained offline handwriting recogniser that takes raw pixel data as input.	Multi-dimensional recurrent neural networks	91.4% accuracy on the Arabic Handwriting Recognition

10.	2014	Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks	ICLR	In this paper, author propose a unified approach that integrates the three steps i.e. localization, segmentation, and recognition via the use of a deep convolutional neural network that operates directly on the image pixels.They employ the DistBelief implementation of deep neural networks in order to train large, distributed neural networks on high quality images.The performance of this approach increases with the depth of the convolutional network, with the best performance occurring in the deepest architecture with eleven hidden layers.	DistBelief Deep Neural Networks	97.84% accuracy

CHAPTER 4

PROBLEM DEFINITION AND SCOPE

4.1 PROBLEM STATEMENT

Human Vision has always been a topic of intense discussion as to how brain perceives patterns and objects in real life scenarios with utmost ease. Modelling the vision, on the other hand, is pretty difficult to surmise, as it falls short to mimic the complexities of natural neural network. Nowadays, with all the technological advances at its side, artificial deep neural network has become the state-of-the-art solutions for computer vision experts. Our aim is to implement this deep learning technology for pattern detection and recognition.

4.1.1 Goals and objectives

1. To perform analysis of different deep learning techniques for pattern detection and recognition.
2. To implement deep learning algorithm on image dataset for obtaining state-of-the-art performance.
3. To obtain the optimized result by increasing efficiency of deep learning algorithm.

4.1.2 Statement of scope

- The software will be able to localize and recognize the individual right whale in the given input image.

4.2 SOFTWARE CONTEXT

- To make a platform which will help in identifying the individual right whale in aerial photographs.
- To provide a tool which automates the process of right whale recognition.
- To provide an application which can be used to give a computer automated input extracted from image.

4.3 MAJOR CONSTRAINTS

- Image should contain single right whale in it.
- The resolution of image should be good enough for extracting whale head from it. Also there should be minimum distortions in image.
- For recognition right whale head should be aligned in specific direction.

4.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

The complete problem statement can be solved by using divide and conquer approach. The aim is to design a tool for automating the process of right whale recognition. This tool consists of mainly two parts namely localization and classification algorithm. This complete problem can be divided into two subproblems :

- Efficient feature extraction algorithm
- Training of algorithm

These can be solved independently and then merged to form the complete solution to form the tool for recognizing individual right whale from the aerial image.

4.5 SCENARIO IN WHICH MULTI-CORE, EMBEDDED AND DISTRIBUTED COMPUTING USED

For faster training of the learning algorithm multicore GPUs can be used.

4.6 OUTCOME

- Outcome of the project: An automated system for localization and recogntion of right whale from images. This can be further manipulated for conservation of the right whales.

4.7 APPLICATIONS

- Real time face detection for right whales.
- Collisions with ships and entanglement in fishing gear have been identified as major sources of mortality and serious injury to right whales. Hence real time help can be provided to save such right whales.
- This same generalized system can be extended for conservation of other animals such as tigers, etc.

4.8 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement
1	CPU Speed	2 GHz
2	RAM	Pentium(R) i7 CPU @2.00 GHz
3	Hard-disk	500 GB
4	GPU	TitanX

Table 4.1: Hardware Requirements

4.9 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System: Ubuntu 14.04
2. IDE: Jupyter
3. Programming Language: Python with Theano Library

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

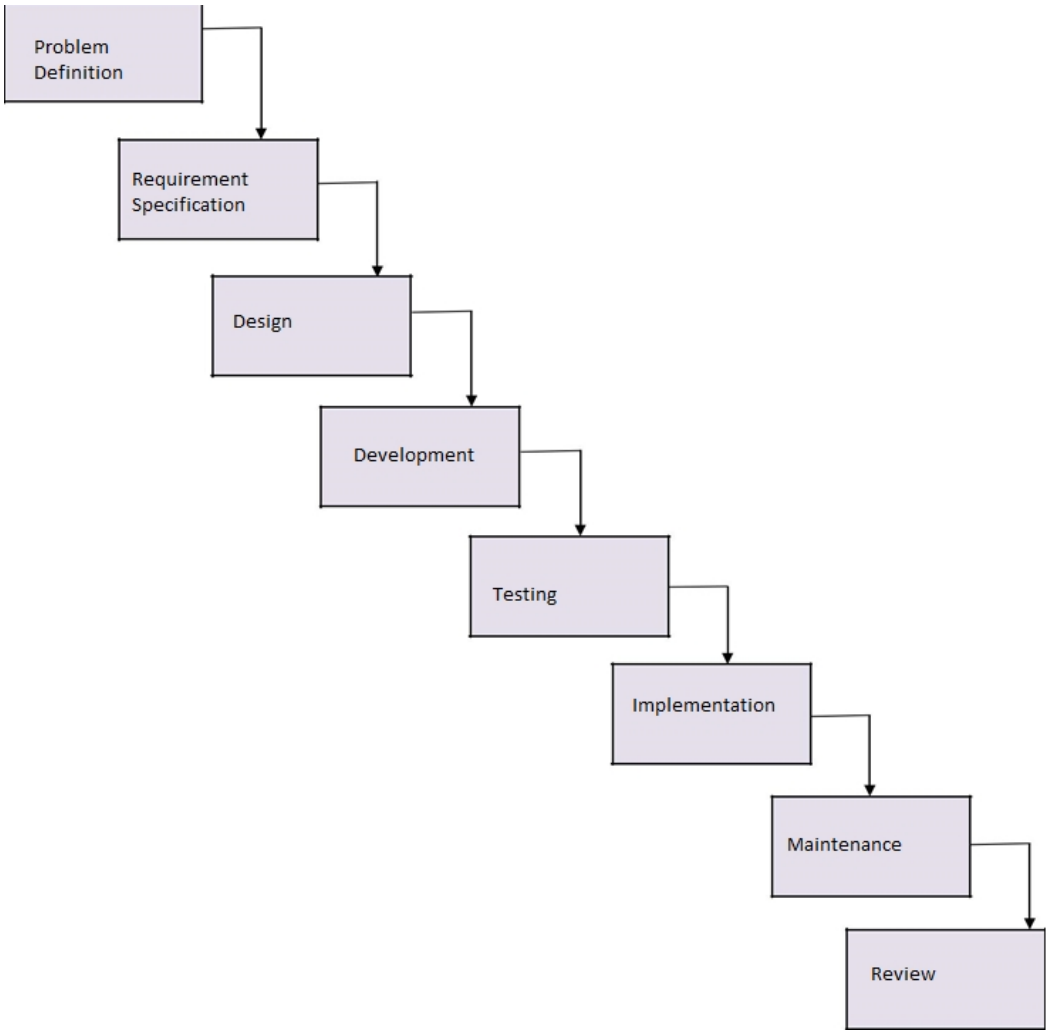


Figure 5.1: Waterfall model

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

- Softwares used are Free Open Source Software Tools

5.1.1.2 Time Estimates

- 3-4 Months

5.1.2 Project Resources

5.1.2.1 Hardware Resources Required

Sr. No.	Parameter	Minimum Requirement
1	CPU Speed	2 GHz
2	RAM	Pentium(R) i5 CPU @2.00 GHz
3	Hard-disk	500 GB
4	GPU	TitanX

Table 5.1: Hardware Requirements

5.1.2.2 Software Resources Required

1. Operating System: Ubuntu 14.04
2. IDE: Jupyter
3. Programming Language: Python with Theano Library

5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

To create the application which can be used to recognize the individual right whale from aerial photographs.

5.2.1 Risk Identification

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

   Yes requirement of this project not only helps the company but also the customers to have a reliable and better platform.

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

   Yes as the project provides effective right whale recognition platform for researchers from aerial photographs and minimizes the human efforts.

3. Are requirements fully understood by the software engineering team and its customers?
   Yes the requirements of project are fully understood by the team as per the user point of view.

4. Have customers been involved fully in the definition of requirements?
   Yes , the dataset of images which can be detected and recognized is made available.

5. Do end-users have realistic expectations?
   Yes real-time working of this project is the basic need and no room for lagging in time basis.

6. Does the software engineering team have the right mix of skills?
   The basic knowledge of unity, programmatic skills ,work management is required and is possessed by each of them.

7. Are project requirements stable?
   Yes the requirements are stable.

8. Is the number of people on the project team adequate to do the job?
   Yes, three people are adequate enough to work on this job to make it successful and even make it more generalize.

9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
   Yes the customers and the developers both are agreed on the requirement list for system and product fulfillment.

5.2.2 Risk Analysis

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

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	The design of efficient algorithm to train the system	Low	Low	High	High
2	Size of application may go on increasing	Low	Low	High	High

Table 5.2: Risk Table

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

Table 5.3: Risk Probability definitions

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

Table 5.4: Risk Impact definitions

Risk ID	1
Risk Description	The dataset for training of the algorithm
Category	Requirement
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	More searching for appropriate dataset will solve this issue.
Risk Status	Occurred

Risk ID	2
Risk Description	The design of efficient algorithm to train the system.
Category	Design and Implementation
Source	Software Design Specification documentation review.
Probability	Medium
Impact	High
Response	Mitigate
Strategy	Better understanding of working of algorithm will solve this problem.
Risk Status	Identified

Risk ID	3
Risk Description	Size of project may go on increasing
Category	Implementation
Source	This was identified during early development and testing.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Better testing and regular check will resolve this issue.
Risk Status	Identified

5.3 PROJECT SCHEDULE

	Task Name	Start	Finish
1	Group Formation	7/1/2015	7/17/2015
2	Guide Allocation	7/17/2015	7/17/2015
3	Ideation Stage	7/17/2015	8/21/2015
4	Understanding the problem	8/21/2015	8/28/2015
5	Reporting the problem stater	8/31/2015	8/31/2015
6	Literature survey	8/3/2015	12/9/2015
7	SRS and High Level Design,	9/10/2015	9/21/2015
8	Semester I Report Submissic	10/5/2015	10/8/2015
9	Basic Installation	11/19/2015	11/25/2015
10	**Understanding and decidi**	**1/1/2016**	**2/1/2016**
11	Coding 1st Module	1/1/2016	2/1/2016
12	Testing and Debugging	2/2/2016	2/15/2016
13	Coding 2nd Module	2/16/2016	3/3/2016
14	Final Report	3/23/2016	3/23/2016

5.3.1 Task network

Sr. No	Task ID	Name Of Task
1.	T-1	Study of Image feature extraction methods, Text detection and recognition method, and other decision related concept.
2.	T-2	Preliminary design of solution.
3.	T-3	Refined design of the solution.
4.	T-4	Preparation for the first prototype.
5.	T-5	Revision of the prototype.
6.	T-6	Testing and Documentation.

5.3.2 Timeline Chart

(Refer Annexure C)

5.4 TEAM ORGANIZATION

The manner in which staff is organized and the mechanisms for reporting are noted.

5.4.1 Team structure

5.4.2 Management reporting and communication

Conceptual reading in the form of reading from papers. The problem definition and its details were contributed and discussed by all of us. A weekly meeting with the internal guide is conducted and its record is maintained. This may be in the form of direct interaction or through email. Communication between team members is direct as well as through email.

Role	Responsibilities	Participants
Project Guide	Manage project in accordance to the project plan. -Direct/lead team members Toward project objectives - Oversee discussions related to design, performance evolution.	Prof. D.T.Mane, Computer engineering, PICT, Pune.
Project Participants	Communicate project goals, Status and progress throughout the project. Review and approve project deliverables. Coordinates participation of work groups, individuals and stakeholders assure quality of products that will meet the project objectives. Identify risks and issues and help in resolution.	PICT Group members • Faizan Shaikh • Shraddha Gunjal • Manasi Khapke

Table 5.5: Team Structure

CHAPTER 6

SOFTWARE REQUIREMENT SPECIFICATION

6.1 INTRODUCTION

6.1.1 Purpose and Scope of Document

The purpose of software requirement specification document is to provide detailed overview of our software product, its parameters and goals. This document provides projects user interface, hardware and software requirements. It defines how our clients, Team and audience see the product and its functionality. It helps any designer and developer to assist in software delivery lifecycle processes. This document provides detailed description of software context, its usage scenario, data model and description, data objects and their relationships, functional model and description, performance issues, software interfacing description, limitations and validation criteria.

6.1.2 Overview of responsibilities of Developer

- Perform project design and development activities according to customer specifications.
- Work with team in developing project plan, budget and schedule.
- Coordinate with team in preparing project documents.
- Track project progress regularly and develop status report.
- Ensure that project is completed within allotted budget and timelines.
- Research and recommend new technologies to carry out project development tasks.
- Provide assistance to other Developers, perform reviews and provide feedback for improvements.
- Develop cost reduction initiatives while maintaining quality and productivity.

6.2 USAGE SCENARIO

6.2.1 User profiles

1. **Trainer**: Trainer provides an input which will help to train system.
2. **User**: User performs an action.
3. **Datasets**: Training images will be from datasets.
4. **Application**: Application performs actions like taking image input from the user and displaying its text content.

6.2.2 Use-cases

Sr No.	Use Case	Description	Actors	Assumptions
1	Upload Image	The input image for testing should be given.	User	The image should contain right whale in it to recognize.
2	Initialize	The system would be initialized so as to take image as input and used for further processing.	System	The parameters for initializing the system should be identified correctly.
3	Pre-processing	The pre-processing on image should be done by system.	System	The pre-processing conditions and parameters should be given carefully. This should be able to do the functions as resizing and also normalize it.
4	Recognize	The system should be able to detect and recognize right whale from image and give individual's name as an output.	System	The system should be able to recognize and display it correctly.

Table 6.1: Use Cases

6.2.3 Use Case View

Use Case Diagram. Example is given below

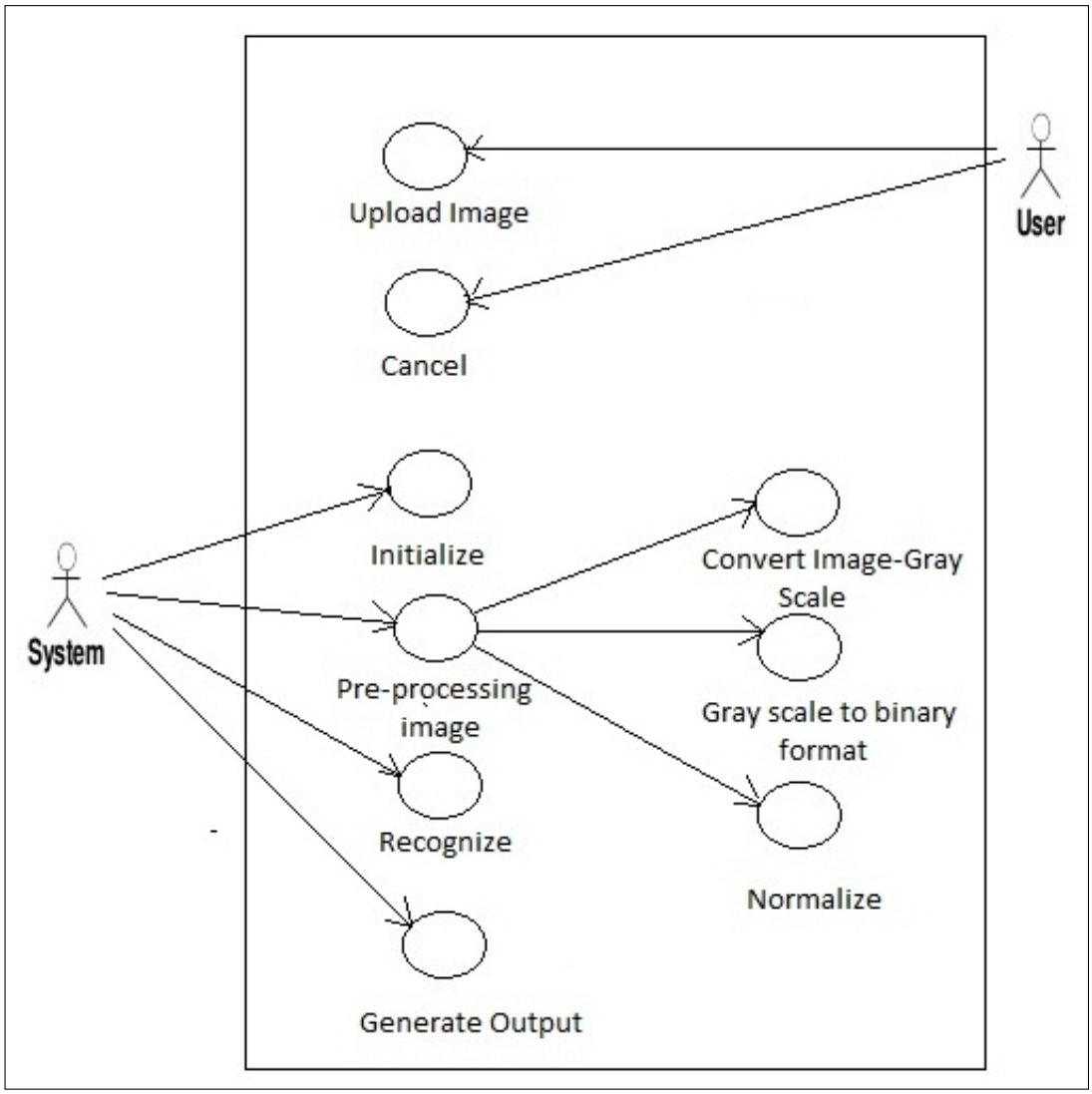


Figure 6.1: Use case diagram

6.3 DATA MODEL AND DESCRIPTION

6.3.1 Data Description

- Right Whale Dataset
  - Right Whale Dataset is collected from NOAA(National Oceanic and Atmospheric Administration) Fisheries.

- Each image from dataset contains a single Right whale. These images were taken over the course of 10 years and hundreds of helicopter trips, and have been selected and labeled by NOAA scientists with their whale IDs.
- The dataset is divided into two parts:
  1. Training dataset with 4544 images.
  2. Testing dataset with 6925 images.
- The number of images per whales varies hugely, as can be seen from the below histogram. There were 24 whales that came with only 1 image in the dataset.

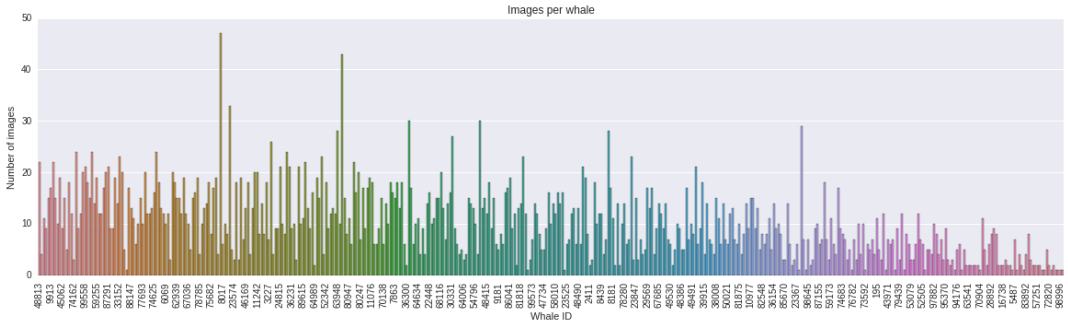


Figure 6.2: Dataset Distribution

6.4 FUNCTIONAL MODEL AND DESCRIPTION

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

6.4.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the ”flow” of data through an information system, modelling its process aspects.

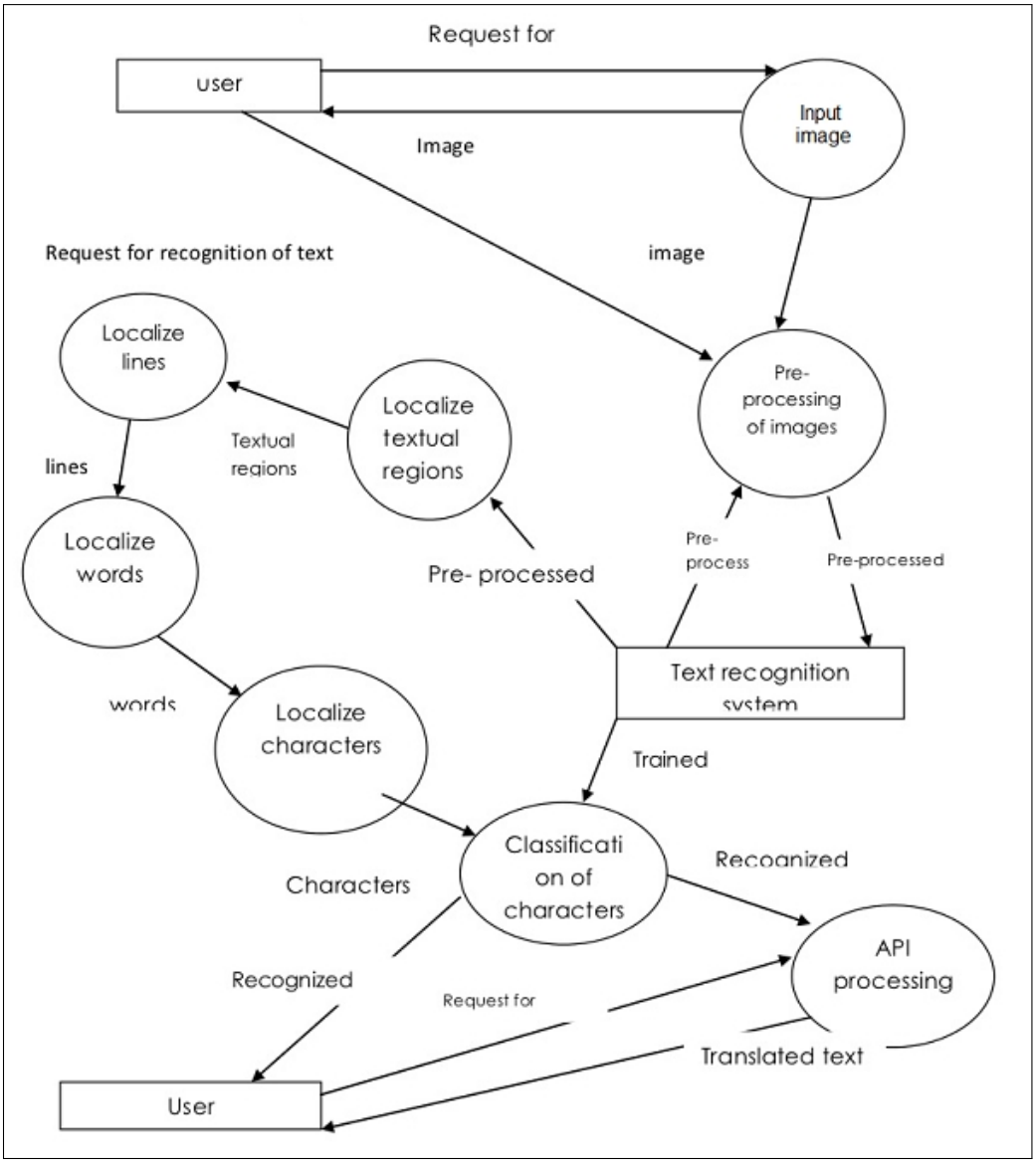


Figure 6.3: Data Flow diagram

6.4.2 Description of functions

A description of each software function is presented. A processing narrative for functions is presented.

6.4.3 Activity Diagram:

- The Activity diagram represents the steps taken.

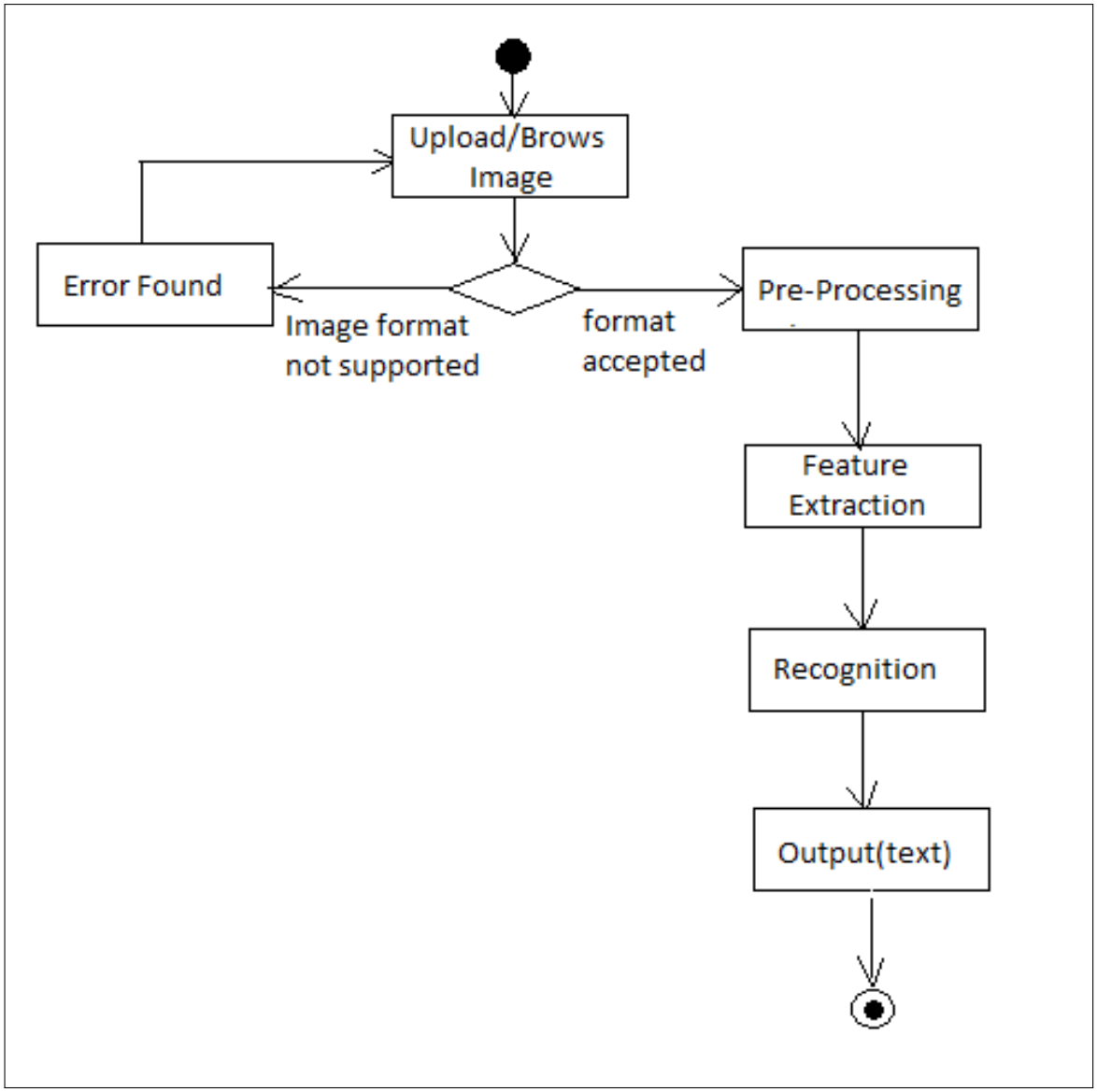


Figure 6.4: Activity diagram

6.4.4 Non Functional Requirements:

- Interface Requirements
  - **External machine interfaces** :

    GPUs if required for training.
  - **External system interfaces**

    Theano Library will be used as external system interface.
  - **Human interface**

    An overview of any human interfaces to be designed for the computer is the common device used everywhere.

- Performance Requirements
  - Restrictions : Image should have higher resolution and less distortions.
  - Limitation: Image should contain only one right whale.
  - Constraints: Image should be captured in sufficient light and right whale should be there in the captured image.

6.4.5 State Diagram:

State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented as a series of events that can occur in one or more possible states

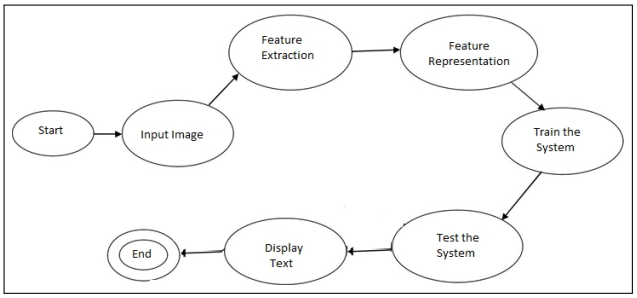


Figure 6.5: State transition diagram

CHAPTER 7

DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 INTRODUCTION

1. Purpose:

   The Software Design Documentation will provide a detailed description of the system to allow for software development to proceed with an understanding of what is to be built. The high level design provides information necessary to describe the details of the software and the system to be developed.

2. Scope:

   This Software Design Documentation is a base level system which will prove the feasibility of building an advanced system that will provide an Optimal route according to users preferences and traffic conditions. The document will sketch a detailed picture of the critical factors that will need consideration.

7.2 ARCHITECTURAL DESIGN

- The system has four units. For training input images are taken from right whale dataset. During preprocessing image resizing is done. Preprocessed image is given to localizer for locating whale head in whole image. Then the localized whale head is cropped and aligned. Next module classifier takes cropped whale head as an input and classifies it into different 447 individuals.

7.3 DATA DESIGN (USING APPENDICES A AND B)

7.3.1 Internal software data structure

To store all the information about the images use files. To store the attributes of images arrays are used.

7.3.2 Global data structure

Files are used as global data structure.

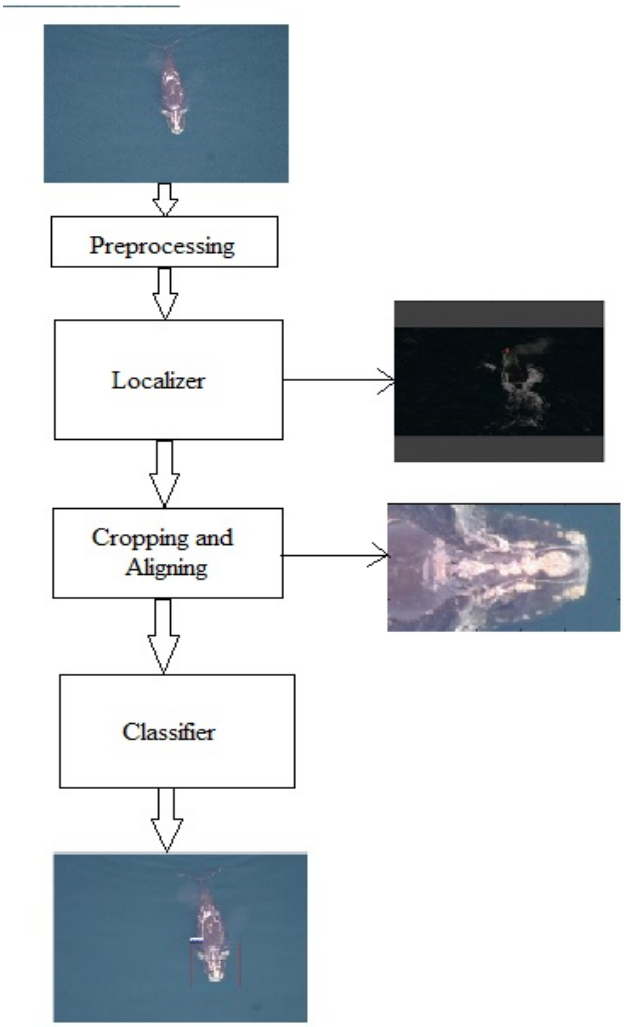


Figure 7.1: Architecture diagram

7.3.3 Temporary data structure

Files created for interim use are described.

7.3.4 Database description

- Right Whale Dataset
  - Right Whale Dataset is collected from NOAA(National Oceanic and Atmospheric Administration) Fisheries.
  - Each image from dataset contains a single Right whale. These images were taken over the course of 10 years and hundreds of helicopter trips, and have been selected and labeled by NOAA scientists with their right

whale IDs.

- The dataset is divided into two parts:
  1. Training dataset with 4544 images.
  2. Testing dataset with 6925 images.
- The number of images per whales varies hugely, as can be seen from the below histogram. There were 24 whales that came with only 1 image in the dataset.

7.4 COMPONENT DESIGN

7.4.1 Class Diagram

The class diagram gives the description of various classes for the object oriented system. It describes the classes of the system as follows:

7.4.2 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It also shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

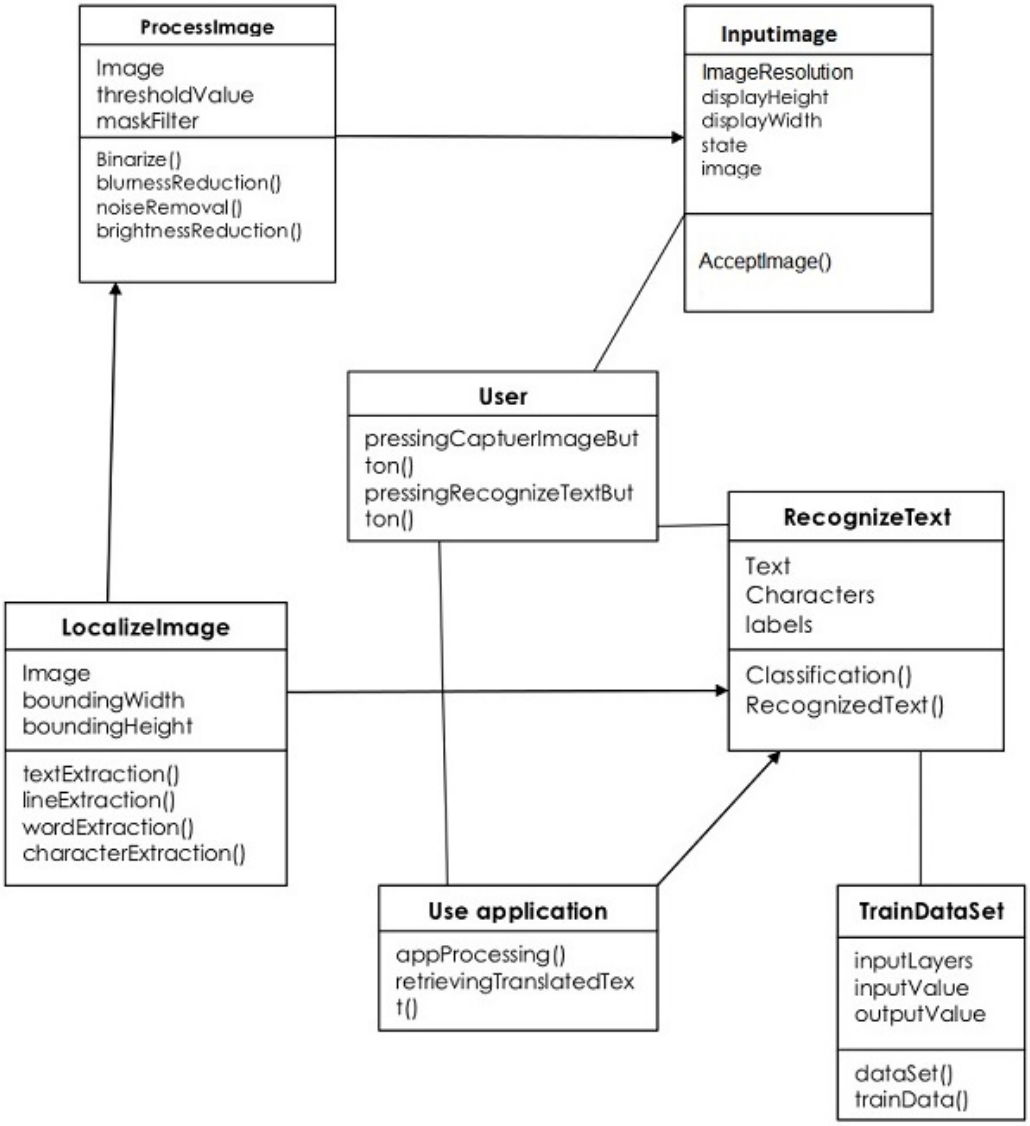


Figure 7.2: Class Diagram

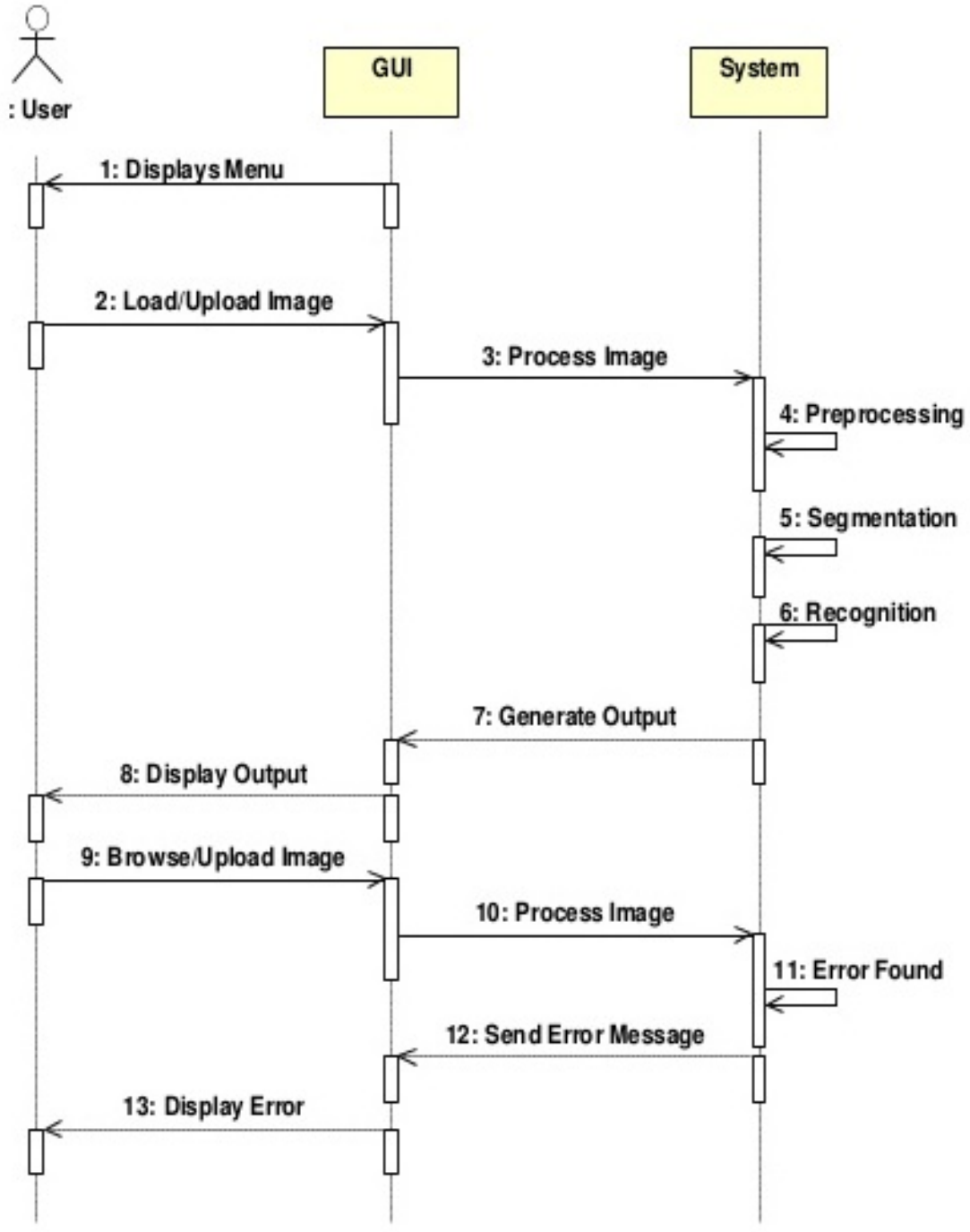


Figure 7.3: Sequence Diagram

CHAPTER 8

PROJECT IMPLEMENTATION

8.1 INTRODUCTION

- Pattern detection and recognition is the act of taking in raw data and taking an action based on the identified category of the pattern.
- Pattern in an image is significant from a computer vision aspect as it opens up a new dimension for human computer interaction.
- While working with natural images, tremendous difficulties such as occlusion, background foreground variation, distortion and even intra-class dissimilarities of patterns need to be handled. Due to these difficulties, traditional methods fail to give more accurate results on classification and other tasks.
- To cope with these problems deep learning techniques could be useful as they prove to be remarkable in feature discrimination. They separate noise and information from actual data.
- A special type of deep learning technique called convolutional neural network (CNN) is becoming a key tool for computer vision research.
- To replace the complex preprocessing pipeline Convolutional Neural Network (CNN) can be used. CNN is a feed- forward multi-layer perceptron that works as a hierarchical feature extractor for transforming raw data into information.
- We focus on using CNN for individual right whale localization and recognition from aerial photographs.

8.2 TOOLS AND TECHNOLOGIES USED

- IDE:Jupyter
- Libraries uesd:
    1. Theano: It is a python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It is ossible to attain speeds rivalling handcrafted implementations for problems involving large amounts of data.

2. Lasagne: It is a lightweight library to build and train neural networks in Theano. It is possible to write Python code and execute the training on nVidia GPUs with automatically generated CUDA code.

3. scipy: It is a collection of mathematical algorithms and convenience functions built on the Numpy extension of Python. It provides the user with high-level commands and classes for manipulating and visualizing data.

4. Sklearn: It is open source machine learning library for various classification, regression and clustering algorithms and to interoperate with the Python numerical and scientific libraries NumPy and Scipy.

8.3 ALGORITHM DETAILS

8.3.1 Convolutional Neural Network

- Convolutional neural network is a feed-forward multilayer perceptron. It is inspired by the way biological nervous system works and processes information inside the brain. Instead of the predefined and handcrafted features, it uses the extracted features for classification and other tasks.

- A neural network is generally organized into multiple layers viz. the input layer, the hidden layer, and the output layer consisting of numerous units or neurons. It also has a set of edges between adjacent layers for connecting neurons. Each neuron is connected to every neuron from the preceding layer in a feedforward manner.

- The activations of connected neurons are computed layer by layer and last layer activations are the predictions of the network. More formally let x i be an input, the activation of the hidden layer z j is computed as: where h(x) is an activation function and w ji are the input-to-hidden weights. There is also a bias term b i which helps in regularizing the network. The output y k for the network is computed by:

  where g(x) is an activation function, which is not necessarily the same as h(x),

and w kj are the hidden-to-output weights. The weights w ij and w kj are learned by training on labelled data with backpropagation.

- Backpropagation minimizes an error criterion E(w). To minimize this criterion the gradient of the function E(w) with respect to w is computed, which is then used in an optimization algorithm such as Stochastic Gradient Descent (SGD) to update the weights. Gradients are computed layer by layer starting from the output layer. First, the gradients from the output layer are backpropagated to the neurons of the hidden layer. Hidden Layer neurons accumulate all gradients from outgoing connections and propagate the error back to the inputs of these neurons.

- CNN has layers with shared weights. Each neuron in layer li+1 is connected with neurons in layer li and the weights between these layers are shared. To predict the activations for the next layer a convolution operation on the inputs of the layer is applied. Convolutional layers are followed by a subsampling operation, where responses are pooled in a local receptive field to a single value by computing mean, max, Lp-pooling, or sampling. Dropout is a regularization method proposed by Hinton et al. [23] for regularizing ANNs by randomly setting inputs during training to 0. This will help in reducing the overfitting problem in the network.

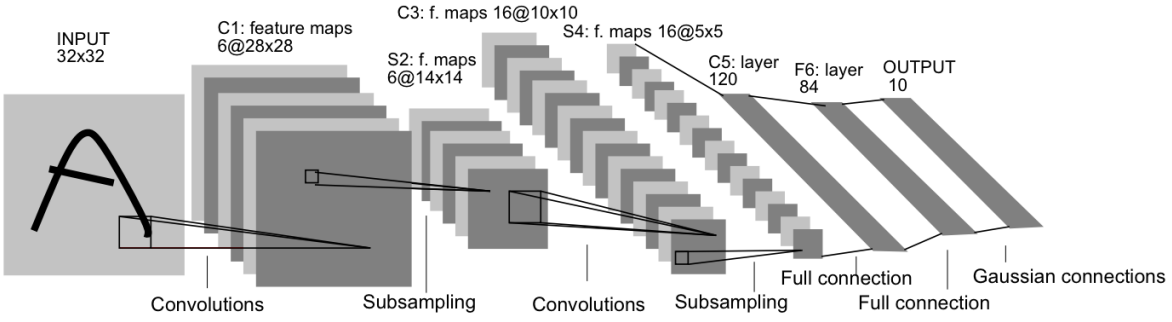


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Figure 8.1: Basic CNN Architecture D

- Our problem can be divided into Localization and Classification task. For each task different CNN architecture is built. In localization, the whale head is detected from entire image. Head localizer is the regression-based problem and gives output as two numbers which represents blow-hole and bonnet point for whale head in whole image. This localized head part is given to CNN classifier to extract the features and classify them into different 447 categories of right whales.

8.3.1.1 CNN Localizer

8.3.1.2 CNN Classifier

8.4 VERIFICATION AND VALIDATION FOR ACCEPTANCE

CHAPTER 9

SOFTWARE TESTING

9.1 INTRODUCTION

In this document we are describing different testing strategy for our proposed system.

9.1.1 Goals and objectives

The main goal is validating the input image sequence and proper processing of given input which will help in smooth running of proposed system.

9.1.2 Statement of scope

In proposed system, we are recognizing the right whale from aerial photographs. The image given as input consists of single right whale. Our project can detect the right whales only, not other objects in the image, so this may act as an invalid input and not incorrect.

9.2 TYPE OF TESTING USED

9.2.1 Unit testing

Unit Testing is the first level of testing. As an individual module is being written, testing will be conducted for each and every piece of functional code by compiling, reviewing and fixing compiler errors. When the coding stage of the module is complete, the module will then be tested according to its purpose. In this stage, the module will be tested with all reasonable and expected inputs to see if it does its prescribed task. Finally, the module will be tested for security issues.

- Module I: Localizing whale head.
  Input: Image of right whale.
  Output: Bounding box around whale head.

- Module II: Recognize individual whale.
  Input: Cropped and aligned whale head.
  Output: Recognized whale name.

9.2.2 Integration testing

After all individual modules have been unit tested, and are properly functioning, they will be put together to form a program. This program will then be tested for general errors including compiler and runtime errors, input errors, and efficiency. Any errors will be corrected at this stage. If the program is inefficient in some computation, the cause of the inefficiency will be isolated and fixed, whether it be at the unit level or the integration level.

9.2.3 Validation testing

As the program is starting to take shape, lead programmers will make sure it adheres to customer guidelines. If any guideline cannot be followed exactly, then the program will be modified to suit the nearest reasonable alternative.

9.2.4 GUI Testing

GUI testing in proposed system is done on the slideshow of the tested images which shows bounding box around whale head with its name.

9.2.5 High-order testing

Proposed system will be system tested to ensure that all programming packages integrate with each other as intended, and that the final software is integrated well into the native environment. Alpha testing will follow, where the software interface will be delivered to a few people to test. The alpha testers will use the software as it was intended, to make sure no problems exist. Here any issues with the interface or computation will be fixed. The software will then be tested for performance. During performance testing we will test the runtime performance of the software and make sure that the interface performs well in a variety of conditions.

Test case	Input	Expected Output
Valid input image format	Selection of proper input image format, i.e., .png	Recognition of the individual whale according to the selection of image.
Invalid input image format	Selection of improper input image format	No bounding box is created for image.

Table 9.1: Unit Testing

9.3 TEST CASES AND TEST RESULTS

9.3.1 Unit Testing

9.3.2 Integration Testing

Test case	Input	Expected Output
Proper file input to recognize function	.csv file created by localizer module should be given as input to the recognize function	Proper matching of .csv files and recognition of corresponding individual whale.
Invalid input image format	Selection of improper input image format	No bounding box is created for image.

Table 9.2: Integration Testing

9.3.3 Validation Testing

Test case	Input	Expected Output
Valid input image format	Selection of proper input image format, i.e., .png	Recognition of the individual whale according to the selection of image .
Invalid input image format	Selection of improper input image format	No bounding box is created for image.
Valid input image type	Selection of whale images as input image	Recognition of the right whale according to the selection of image.

Table 9.3: Validation Testing

9.3.3.1 Pass/fail criterion for all validation tests

The pass/fail criterion should base on the performance that according to the expected results, showed above.

9.3.4 High-order Testing

9.3.4.1 Performance testing

Performance testing will be done by giving different test samples for same right whale.

9.3.4.2 Alpha testing

Alpha testing will be based on the software developers. Members of the team will each go through the system thoroughly and use the analysis functions on a testing database to ensure quality and correctness.

CHAPTER 10

RESULTS

10.1 SCREEN SHOTS

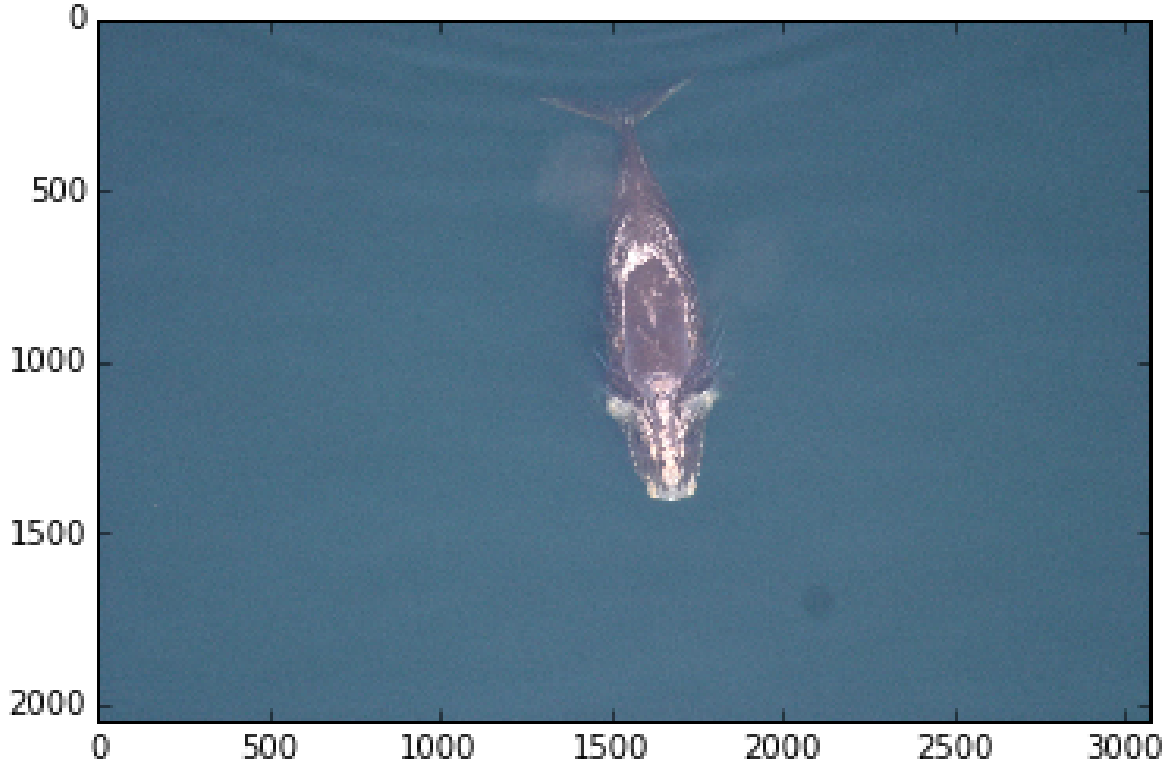


Figure 10.1: Input Whale Image

10.2 OUTPUTS

10.2.1 Outputs / Snap shots of the results

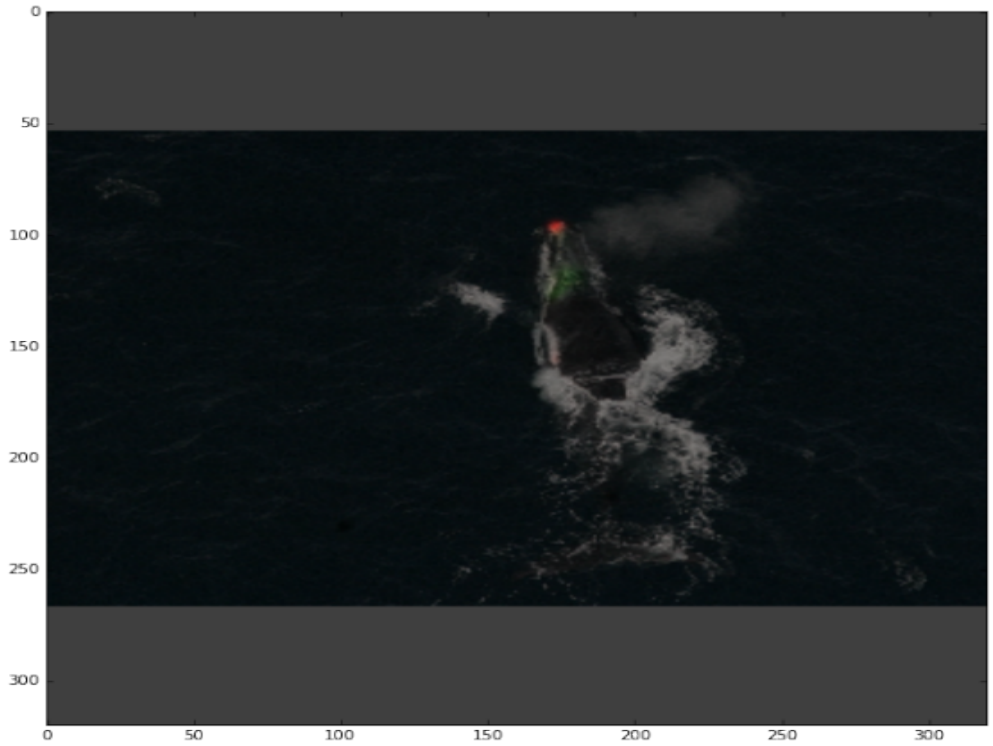


Figure 10.2: Localizer Output

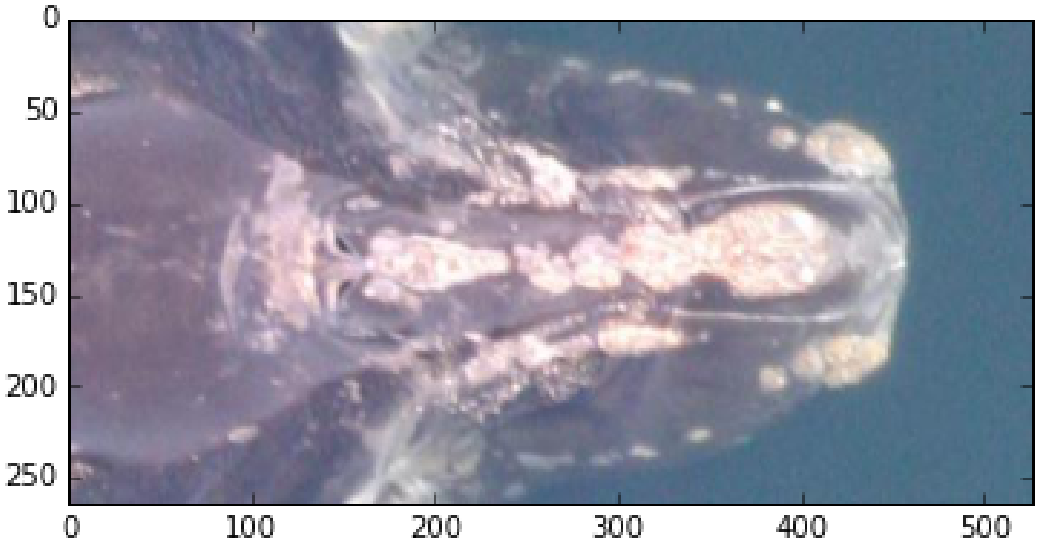


Figure 10.3: Classifier Output

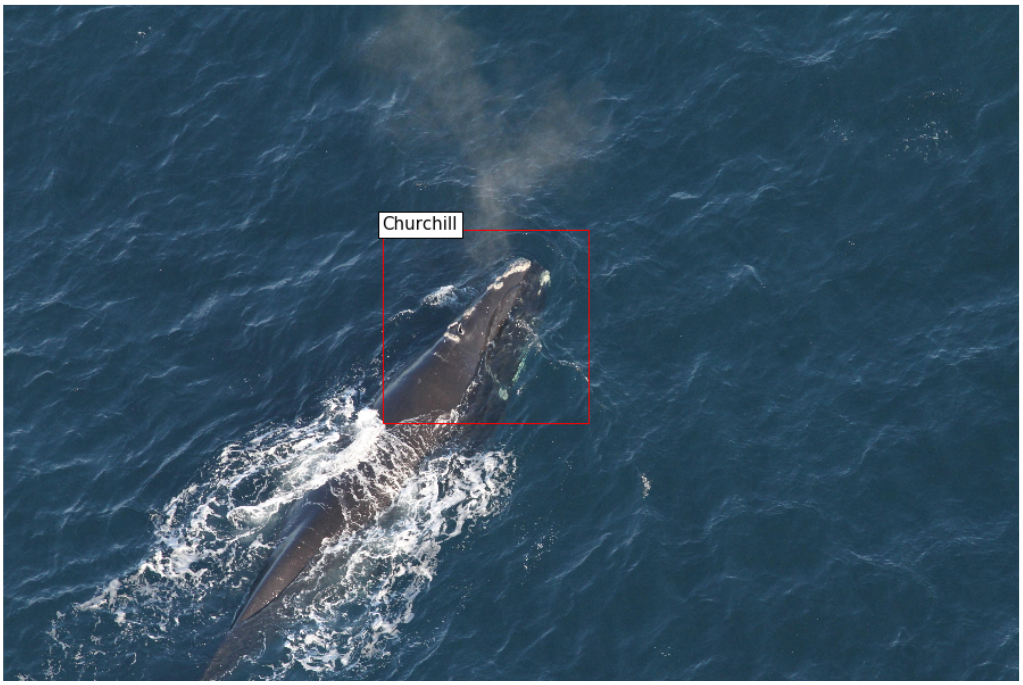


Figure 10.4: Output 1

Figure 10.5: Output 2

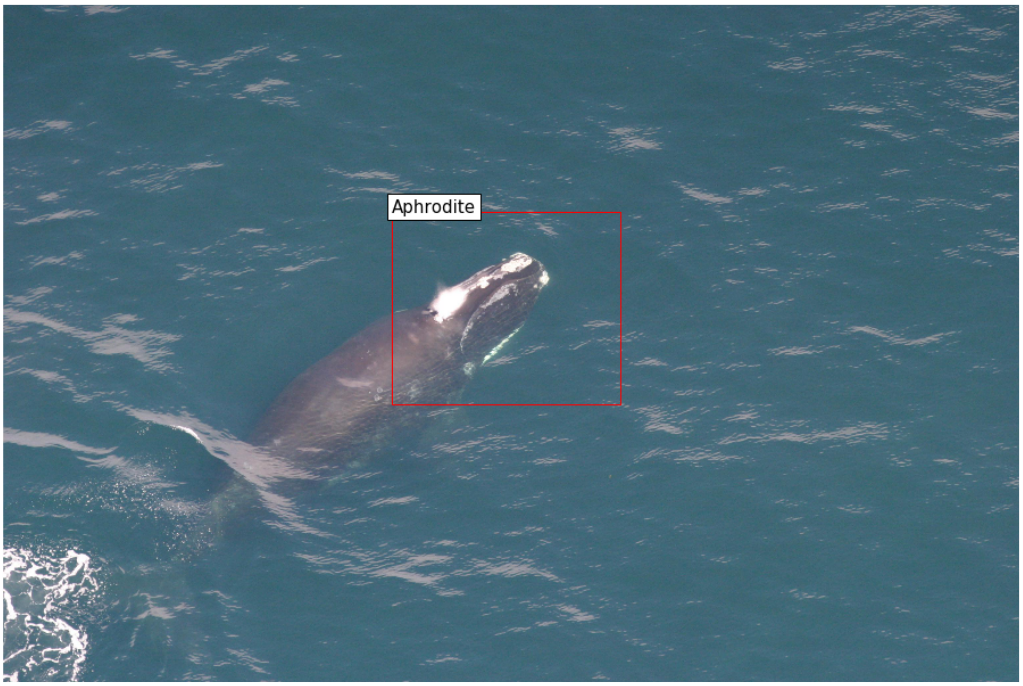


Figure 10.6: Output 3

CHAPTER 11

DEPLOYMENT AND MAINTENANCE

11.1 INSTALLATION AND UN-INSTALLATION

11.2 USER HELP

CHAPTER 12

CONCLUSION AND FUTURE SCOPE

Here we presented CNN for whale recognition from aerial photographs. In comparison to other algorithms, CNN requires minimal pre-processing as it works in a hierarchical manner to extract relevant features. We show that on evaluation, CNN proves to be a state-of-the-art architecture for whale recognition problem.

ANNEXURE A

REFERENCES

(Strictly in ACM Format)

ANNEXURE B

LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN

- To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

  Refer [**?**] for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

I	D	E	A
Increase	Drive	Educate	Accelerate
Improve	Deliver	Evaluate	Associate
Ignore	Decrease	Eliminate	Avoid

Table B.1: IDEA Matrix

- Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

- input x,output y, y=f(x)

ANNEXURE C

LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

It should include assignments such as

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading

- Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.

- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.

- Additional assignments by the guide. If project type as Entreprenaur, Refer [**?**],[**?**],[**?**], [**?**]

ANNEXURE D

PROJECT PLANNER

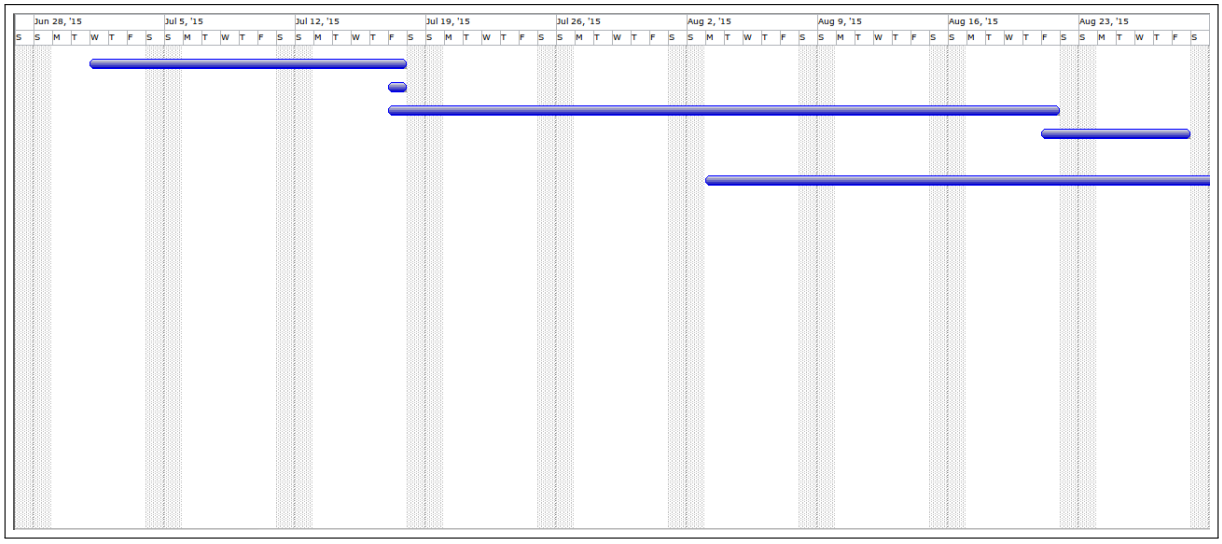


Figure D.1: Gantt Chart

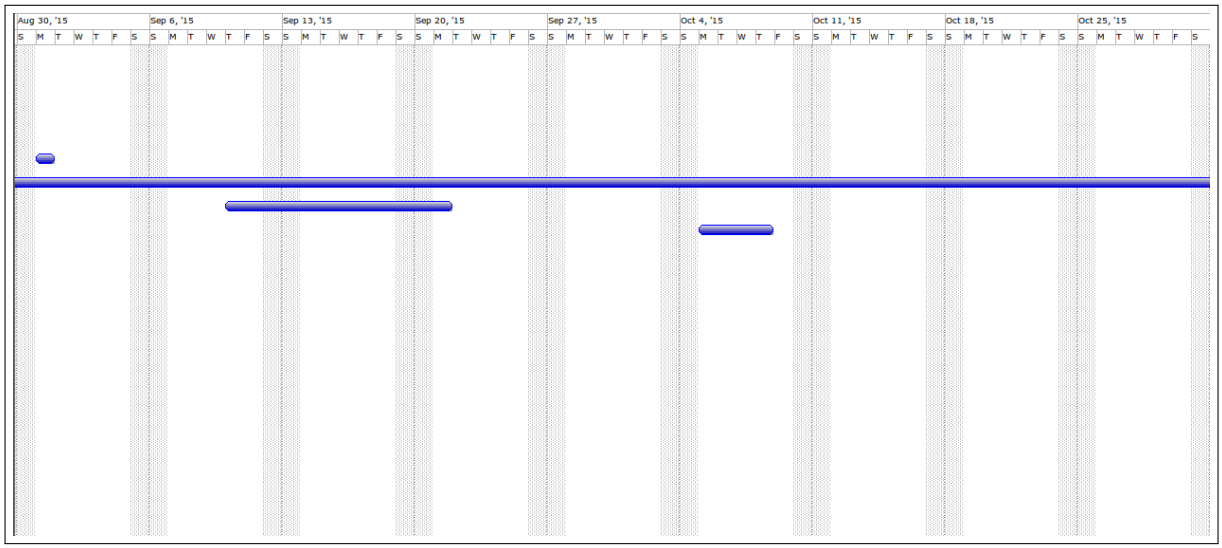


Figure D.2: Gantt Chart

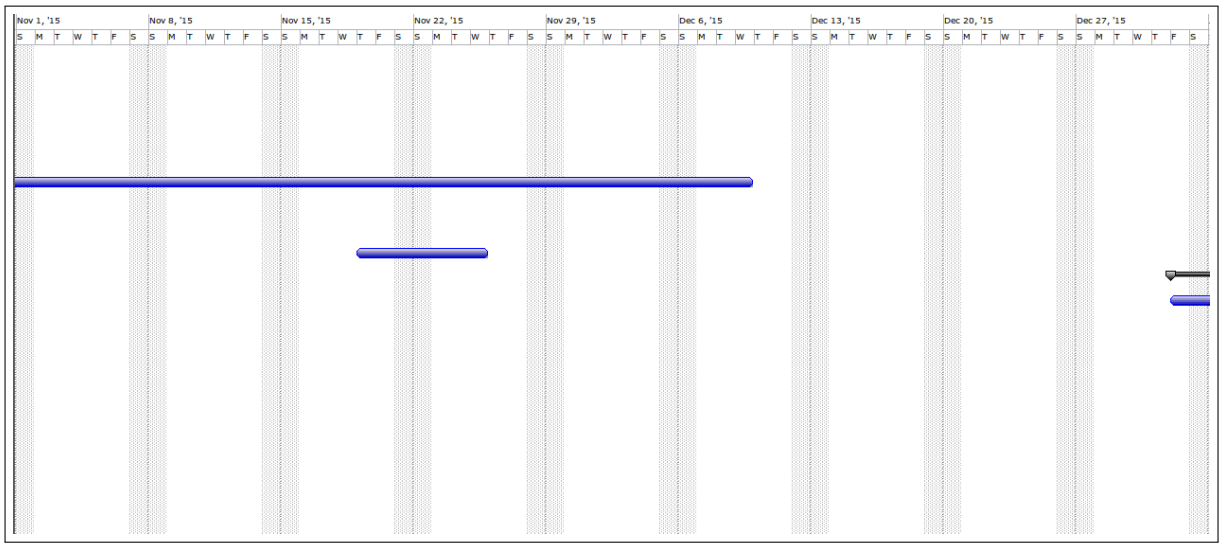


Figure D.3: Gantt Chart

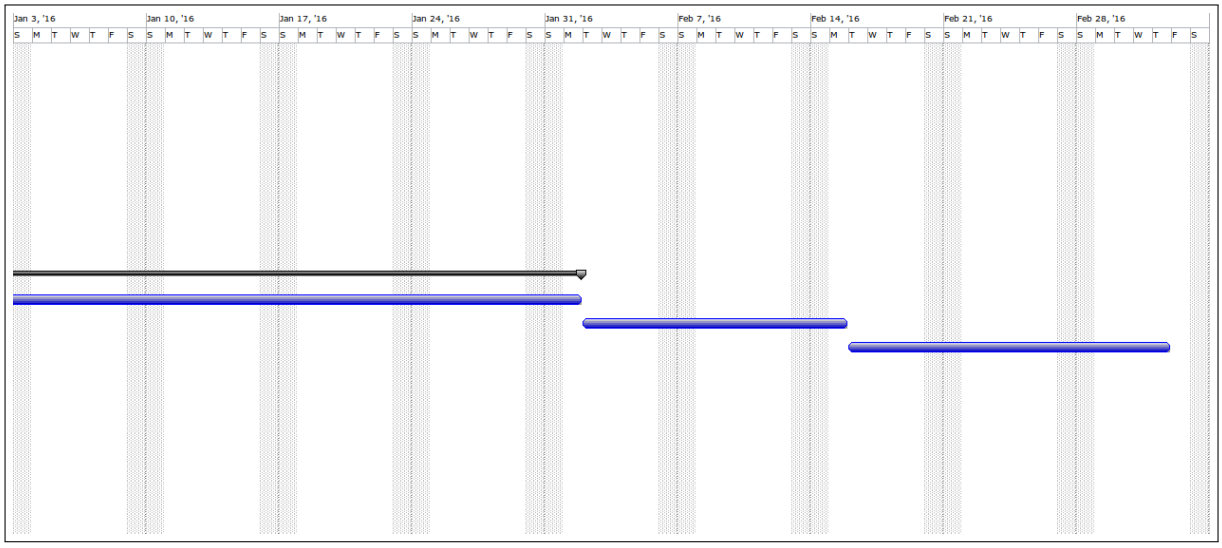


Figure D.4: Gantt Chart