A PRELIMINARY PROJECT REPORT ON

INPAINTING OF DEGRADED DOCUMENT IMAGE USING DEEP NEURAL NETWORK

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Sayali Mahajan	Exam No: B150134308
Kranti Pawar	Exam No: B150134344
Sonali Wagh	Exam No: B150134347
Ankush Jadhav	Exam No: B150134333
Himanshu Shekhar	Exam No: B150134337

DEPARTMENT OF COMPUTER ENGINEERING

K. K. Wagh Institute Of Engineering Education & Research

Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik-422003

SAVITRIBAI PHULE PUNE UNIVERSITY

2018-19

SAVITRIBAI PHULE PUNE UNIVERSITY

2018-19

K. K. Wagh Institute Of Engineering Education & Research

CERTIFICATE

This is to certify that the Project Entitled

"BE PROJECT TITLE"

Submitted by

Sayali Mahajan	Exam No:B150134308
Kranti Pawar	Exam No:B150134344
Sonali Wagh	Exam No:B150134347
Ankush Jadhav	Exam No:B150134333
Himanshu Shekhar	Exam No:B150134337

is a bonafide work carried out by Students under the supervision of Prof. Guide Name and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University, for the award of Bachelor of Engineering (Computer Engineering).

Prof. Guide Name	Prof. HOD Name
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

Dr. K. N. Nandurkar

Principal

Place: Nashik

Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on ***'IN-PAINTING OF DEGRADED DOCUMENT IMAGE USING DEEP NEURAL NETWORK'****.*

We would like to take this opportunity to thank our internal guide ***Prof. Smita Patil*** *for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Dr. S.S.Sane****, Head of Computer Engineering Department, CollegeName for his indispensable support, suggestions.*

In the end our special thanks to ***Prof. Anand Kolapkar and Prof. Suruchi Malao*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.*

Sayali Mahajan
Kranti Pawar
Sonali Wagh
Ankush Jadhav
Himanshu Shekhar
(B.E. Computer Engg.)

ABSTRACT

Document Images get degraded due to unbalance illumination spread over document including smearing of text, bleeding of ink to the other side of page, degradation of paper ink due to aging, manuscript characters from background side appear as noise on the lead side and get blend with the lead side characters etc. Binarization is used to recover text from degraded document images. Recovering text from degraded document images is a very difficult task due to inter or intra variation between background and foreground pixels. Then an interpolation inpainting is performed to create the background estimationand foreground (text) estimation, that will be used as feature to train and test a su-pervised artificial neural network. We are going to perform Deep Neural Network (DNN) to learn about the text pixeland background pixels and define the edges of the text pixels

INDEX

1 Introduction **1**

1.1 Motivation 2

1.2 Problem Definition And Objectives 2

1.3 Project Scope and Limitation 2

1.4 Methodologies of Problem Solving 3

2 Literature Survey **4**

3 Software Requirement Specification **10**

3.1 Assumption and Dependencies 11

3.2 Functional Requirement 11

3.3 External Interface Requirements (If Any) 12

3.3.1 User Interfaces 12

3.3.2 Software Interfaces 12

3.4 Non Functional Requirements 12

3.4.1 Performance Requirements 13

3.5 System Requirements 13

3.5.1 Data sets Requirements 13

3.5.2 Software Requirements (Platform Choice) 13

3.6 Analysis Models : SDLC Model applied 14

4 System Design **15**

4.1 System Architecture 16

4.2 Mathematical Model 16

4.3 Data Flow Diagram 17

4.4 UML Diagram 17

5 Project Plan **20**

5.1 Project Estimate 21

5.1.1 Reconciled Estimate 21

5.1.2 Project Resources 21

5.2 Risk Management 22

5.2.1 Risk Identification 22

5.2.2 Risk Analysis 22

5.2.3 Overview of Risk Mitigation, Monitoring, Management . . 22

5.3 Project Schedule 22

5.3.1 Project Task Set 22

5.3.2 Task Network 24

5.3.3 Timeline Chart 24

5.4 Team Organization 24

5.4.1 Team Structure 24

6 Project Implementation **26**

6.1 Overview of Project Modules 27

6.2 Tools and Technology used 27

6.3 Algorithm Details 27

6.3.1 Otsu's Algorithm 28

6.3.2 Sauvola's Method 28

6.3.3 Niblack's Algorithm 29

6.3.4 How Convolutional Neural Network work? 29

7 Software Testing **31**

7.1 Types of Testing 32

8 Results **33**

8.1 Outcomes 34

8.2 Screen Shots 34

9 Conclusions **35**

9.1 Conclusion 36

9.2 Future Work 36

9.3 Applications 36

Annexure A **37**

Annexure B Plagiarism Report **39**

Annexure C References **41**

List of Figures

3.1 Graphical User Interface of the system 12
3.2 Software Development Life Cycle 14

4.1 Flow chart for the system 16
4.2 Flow chart for the system 18
4.3 Basic Neural Network 19
4.4 Fully loaded Neural Network 19

5.1 Risk Analysis 23
5.2 Task Network 25

6.1 Otsu's Method 28
6.2 Comparision between different algorithms 29

8.1 First screen short of the system. 34
8.2 System with other images. 34

List of Tables

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

There are different motivations for this project, all related to image processing and machine learning domain. The proposed application will be used in different archives where there is need for document preservation.
Our research is based on the Deep Neural Network and Convolutional Neural Network which is one part of Artificial Network. It is going to have multiple hidden layer, each layer filtering multi-degradation of an image. At Every stage we are removing & adding new layer to remove the degradation in the original image.

1.2 PROBLEM DEFINITION AND OBJECTIVES

The goal of the document image binarization is to create a perfect separation between text pixels and background pixels, The binarization is a type of classification problem that segment the document image into two classes: 0 or 1.
Unfortunately, most of the historical document images suffer from different types of degradation, such as bleed-through, large stains, tears ink, blur. Figure (1) shows some examples of degradation. Those multiple degradation make the process of binarization a challenging task for researchers. The good results of the binarization step are inescapable for improving the quality of text recognition system and document image analysis. Our work here is mainly focused on the recovering the degraded part of the image or we can say the stained part. In some cases it can be entire image. The method exploit the benefit of the artificial network with an automatic extraction of features. This extraction is done with an inpainting phase in two way: one for erase the text pixels to get the background pattern , and the second one to erase the background pixels to get a text pattern.

1.3 PROJECT SCOPE AND LIMITATION

A good Software Requirement Specification(SRS) defines how an application will interact with system hardware, other programs and human users in a wide variety of real world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from

adverse events are evaluated. Scope of our project deals with degraded region of the degraded document image. Unlike other applications available in the market its scope is limited to only small research purposes. Basically our project is likely to help understand the working idea behind the Inpainting or Binarization of degraded document image with the help of Deep Learning. As all the system has some limitation so does this. First of all limitations based on the input images provided to the system. Secondly the deep learning implementation on the system sometimes causes to change the pixel parts due to the unwanted filling operations.

1.4 METHODOLOGIES OF PROBLEM SOLVING

Our approach to solve the project is very basic i.e. Divide and Conquer. We have divided project into small stages as follows, taking image as input and covert it to its respective grey-scale and simultaneously applying Gaussian filter with varying sigma values. Module two do the Adaptive Threshold part i.e. applying one of available algorithm for Binarization i.e. Otsu's, Sauvolas', etc. Third and fourth part is to apply morphology i.e. performing dilation and erosion operation respectively to get the text pixel class or background pixel class. Then our final module will train and test the same operations on the set of images so that our deep neural network can learn and predict the output image.

CHAPTER 2

LITERATURE SURVEY

- Title: A Binarization Method for Degraded Document Image using Artificial Neural Network and Interpolation Inpainting.

  Author: Naouel Ouafek, Mohamed-Khireddine Kholladi

  Description:The binarization of degraded document image is an important phase in the document analysis to extract text from the original image. For this purpose, we present a new multi- stage approach started by, a preprocessing phase to enhance the contrast of the original image and to eliminate noises. Then an interpolation inpainting is performing to create the background estimation and the foreground (text) estimation, that will be used as feature to train and test a supervised artificial neural network. the final stage is the use of the ANN to classify the pixels of the original image into 0 for black and 1 for white to get the final binary image. The suggested method has been tested and evaluated over DIBCO2009 database, the results show the successful of the proposed approach.

- Title: ”Adaptive document image binarization”, Pattern Recognition.

  Author: J. Sauvola, M. Pietika Kinen

  Description: A new method is presented for adaptive document image binarization, where the page is considered as a collection of sub-components such as text, background and picture. The problems caused by noise, illumination and many source type-related degradations are addressed. Two new algorithms are applied to determine a local threshold for each pixel.

  The performance evaluation of the algorithm utilizes test images with ground-truth, evaluation metrics for binarization of textual and synthetic images, and a weight-based ranking procedure for the final result presentation. The proposed algorithms were tested with images including differerent types of document components and degradations. The results were compared with a number of known techniques in the literature. The bench-marking results show that the method adapts and performs well in each case qualitatively and quantitatively.

- Title: Binarization Techniques for Degraded Document Images

Author: Jyotsna, Shivani Chauhan, Ekta Sharma, Amit Doegar

Description: Document Image binarization is the segmentation of the document into foreground text and background. It is done to obtain the clear images from which text can be retrieved easily. Thresholding is used for the segmentation of the document images. This paper, presents a review on various document image binarization techniques. Evaluation performance metrics used for the evaluation of the binarization techniques are also explained.

Comparison of the performance of the binarization techniques based on the performance metrics like PSNR, F-Measure, NRM and MPM is shown. Performance of the techniques is evaluated on the dataset of DIBCO-2009 and DIBCO-2010.

- Title: Degraded Document Image Binarization Techniques

  Author: Jyoti Rani and Davinder Parkash

  Description: Document Image Binarization is performed in the preprocessing stage for document analysis and it aims to segment the foreground text from the document background. A fast and accurate document image binarization technique is important for the ensuing document image processing tasks such as optical character recognition (OCR) and Document Image Retrieval (DIR). This research area has been studied for decades; many techniques have been reported and applied on different commercial document analysis applications. However, there are still some unsolved problems need to be addressed due to the high inter/intra-variation between the text stroke and the document background across different document images. Image binarization is the method of separation of pixel values into dual collections, black as foreground and white as background. Thresholding has found to be a well-known technique used for binarization of document images. Thresholding is further divide into the global and local thresholding techniques.

- Title: Optimal Parameter Selection Technique for a Neural Network Based Local Thresholding Method

Author: Mohammed J Islam, Majid Ahmadi, Yasser Alginahi

Description: Thresholding of a given image into binary image is a necessary step for most image analysis and recognition techniques. In document recognition application, success of OCR mostly depends on the quality of the thresholded image. Non-uniform illumination, low contrast and complex background make it challenging in this application. In this paper, selection of optimal parameters for Neural Network (NN) based local thresholding approach for grey scale composite document image with non-uniform background is proposed. NN-based local image thresholding technique uses 8 statistical and textural image features to obtain a feature vector for each pixel from a window of size (2n + 1)x(2n + 1), where n 1

. An exhaustive search was conducted on these features and found pixel value, mean and entropy are the optimal features at window size 3x3. To validate these 3 features some non-uniform watermarked document images with known binary document images called base documents are used. Characters were extracted from these watermarked documents using the proposed 3 features. The difference between the thresholded document and base document is the noise. A quantitative measure Peak-Signal-to-Noise ratio (PSNR) is used to measure the noise. In case of unknown base document characters were extracted through the proposed 3 features and used in a commercial OCR to obtain the character recognition rate. The average recognition rate 99.25% and PSNR shows that the proposed 3 features are the optimal compare to the NN-based thresholding technique with different parameters presented in the literature.

- Title: Modeling Adaptive Degraded Document Image Binarization and Optical Character System

  Author: Yahia S. Halabi, Zaid SASA, Faris Hamdan, Khaled Haj Yousef

  Description: This paper presents an enhanced system for degraded old document. The developed system is able to deal with degradations which occur due to shadows, non-uniform illumination, low contrast and noise. The developed system is able to separate the two regions of the document. Different filtering

techniques are used in the de-noising step for the purpose of de-noising and a rough estimation of foreground region and background region. Binarization step is applied by computing an approximate background surface of an original image. Final threshold step is performed by combining the calculated background surface with the pre-processed original image, using a threshold parameter for predefined local window of specific size. Different interpolation techniques are used in the final step to achieve better quality binary image which yield to elimination noises, improve the quality of the text regions and preserve stroke connectivity by filling possible breaks, gaps or holes. The second part of this research deals with optical character recognition,OCR.

The result obtained after pre-processing step (typewritten or printed text, usually captured by scanner) is converted into machine-editable text. In this phase, we initially trained the system (on the known samples of each character) in order to to read a specific font "Intelligent system", then we performed the testing step (converting the image into editable text). The adaptive image will pass through several steps: image analyses for characters, detecting individual symbols, line and character boundary detection, resize character, feature extraction, output computation and finally displaying character representation of the Unicode output (on Microsoft office word application). The proposed system is implemented and tested on actual degraded images. The proposed technique offers good output quality and quite fast.

- Title: A Threshold Selection Method from Gray-Level Histograms
  Author: Nobuyuki Otsu
  Description: A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multithreshold problems. Several experimental results are also presented to support the valid-

ity of the method.

- Title: Image Analysis Using Improved Otsus Thresholding Method
  Author: Himanshu Makkar, Aditya Pundir
  Description: A fresh and new algorithm for retrieval of image is accessible using improved Otsu thresholding techniques in this chapter. The fundamental design work of the new algorithm is that the basic segments of the image are used to retrieve images within a digital library. The histogram derived from threshold. These methods include some knowledge of the distribution of an image, & will result in the less miss- classification. Iso-data algorithm is the iterative process for finding a threshold value. It first segments the image into two regions according to a temporary threshold value chosen .It then calculates the mean value of an image corresponding to the two segmented regions. Calculate the new threshold value & repeat until the threshold value does not change any more. Finally, choose this value for the threshold segmentation. One of the advantages of the algorithm is that, for a given retrieval image, the user can select a query segment with which to perform retrieval, thus it can satisfy different needs from different users.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after having a detailed understanding of the topic and what is going to come out of the project. A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behaviour of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

3.1 ASSUMPTION AND DEPENDENCIES

Some part of the project are developed using third party tools such as Tensorflow, Keras(backend=Tensorflow), which enforce us to assume that layers which are added during the model construction are flawless and our system is compiling them accurately.We assume that you have the basic understanding of the domain topics and terminologies used in this report. Everything related to project is simple but difficult for those who doesn't take interest in this.

All dependencies are python-modules used in the development of the project. This project is being developed in python version 3.5 & above. We are going to use python dependencies module so that our project can run efficiently.

3.2 FUNCTIONAL REQUIREMENT

A description of each software function is presented. A processing narrative for function n is presented. (Steps)/ Activity Diagrams. Shown in Fig. 3.1.

3.3 EXTERNAL INTERFACE REQUIREMENTS (IF ANY)

3.3.1 User Interfaces

The Graphical User Interface of the proposed system is to be developed using Python Tkinter and PyQt5.

The GUI consist of basic button for image input which is to be uploaded in run-time. The output button to show the final output according to different input parameter. These parameters are also taken input from the user.GUI also show what classification is used and the accuracy of the model.

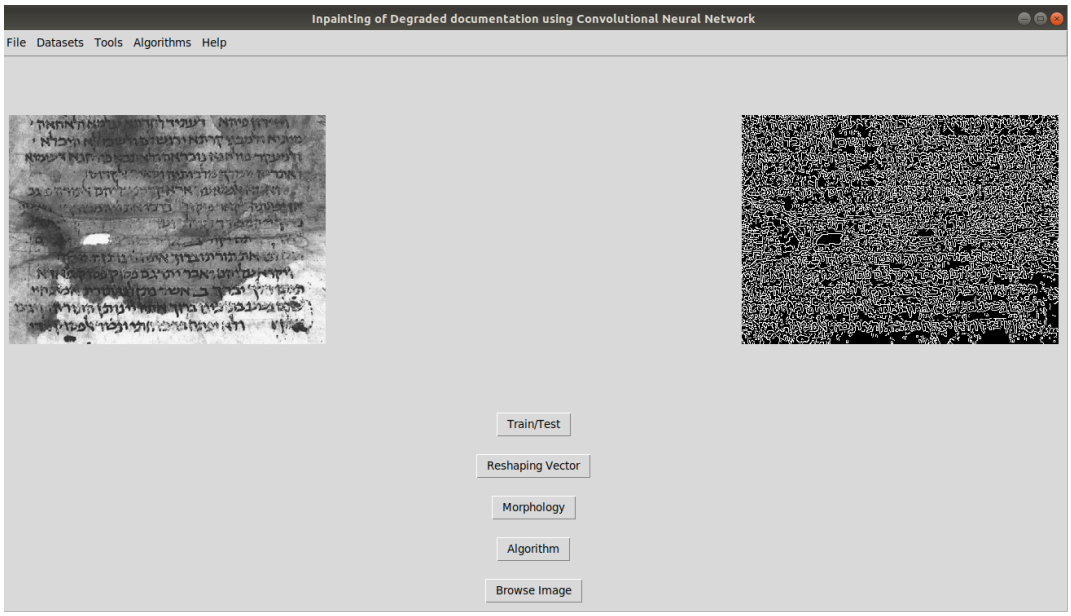

Figure 3.1: Graphical User Interface of the system

3.3.2 Software Interfaces

We have came across a very dynamic and awesome graphical user interfacing tool called Glade , which uses Gtk3+ and python3 to compatible with.

3.4 NON FUNCTIONAL REQUIREMENTS

GUI will be our interface Requirements : Proper classification and training will provide proper output.

Requirements : As our system is fully software oriented we dont need any safety requirement.

3.4.1 Performance Requirements

Since our system is dealing with images and lots of high performance algorithm to work on we will be using Graphics Cards to enhance the system performance and to run the system without any glitch. NVIDIA Graphics card is recommended as they are fast and more reliable when it comes to deep learning.

3.5 SYSTEM REQUIREMENTS

These requirements are used by our team for the system to run efficiently:

- Architecture: x86_64
- Model name: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz
- Operating System: Ubuntu 16.04 & above
- Memory(RAM): 4-Gigabits
- Graphics Card: 2-Gigabits
- Language: Python version 3.5 & above.
- Python Tools: Tkinter, PyQt5, Scipy, Scikit-learn, scikit-image, tensorflow, OpenCV2.

3.5.1 Data sets Requirements

- We will be implementing our project on DIBCO-2009 and DIBCO-2010 data set which is available.
  `http://users.iit.demokritos.gr/~bgat/DIBCO2009/`
  `http://users.iit.demokritos.gr/~bgat/H-DIBCO2010/`
- LRDE Document Binarization Dataset
  `https://www.lrde.epita.fr/wiki/Olena/DatasetDBD`

3.5.2 Software Requirements (Platform Choice)

Our project is entirely implemented and tested on Ubuntu 18.04 LTS.

3.6 ANALYSIS MODELS : SDLC MODEL APPLIED

Agile Software Development Life Cycle models is implemented on the development of the project.In the agile methodology after every development iteration, the customer is able to see the result and understand if he is satisfied with it or he is not. This is one of the advantages of the agile software development life cycle model. One of its disadvantages is that with the absence of defined requirements it is difficult to estimate the resources and development cost.

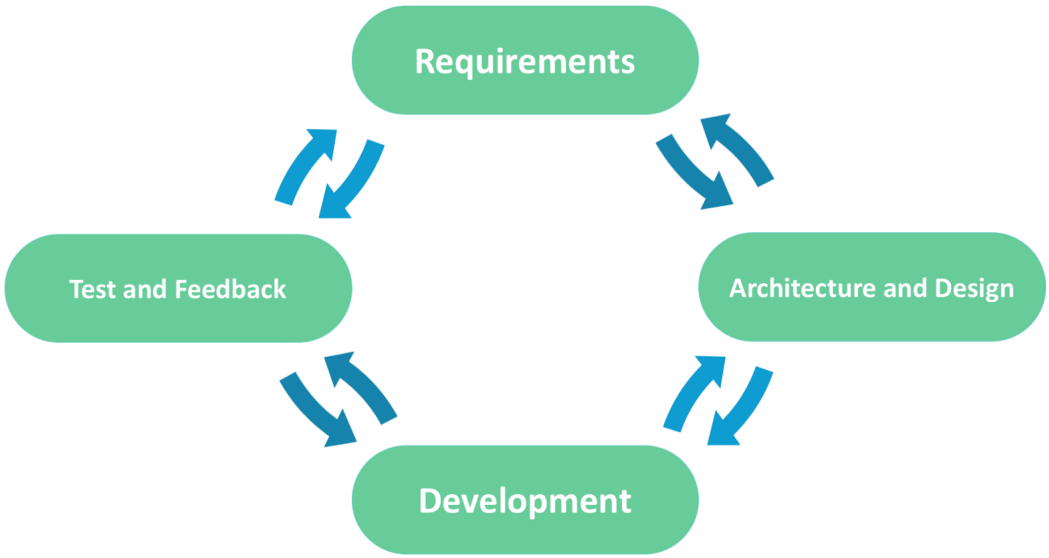

Figure 3.2: Software Development Life Cycle

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

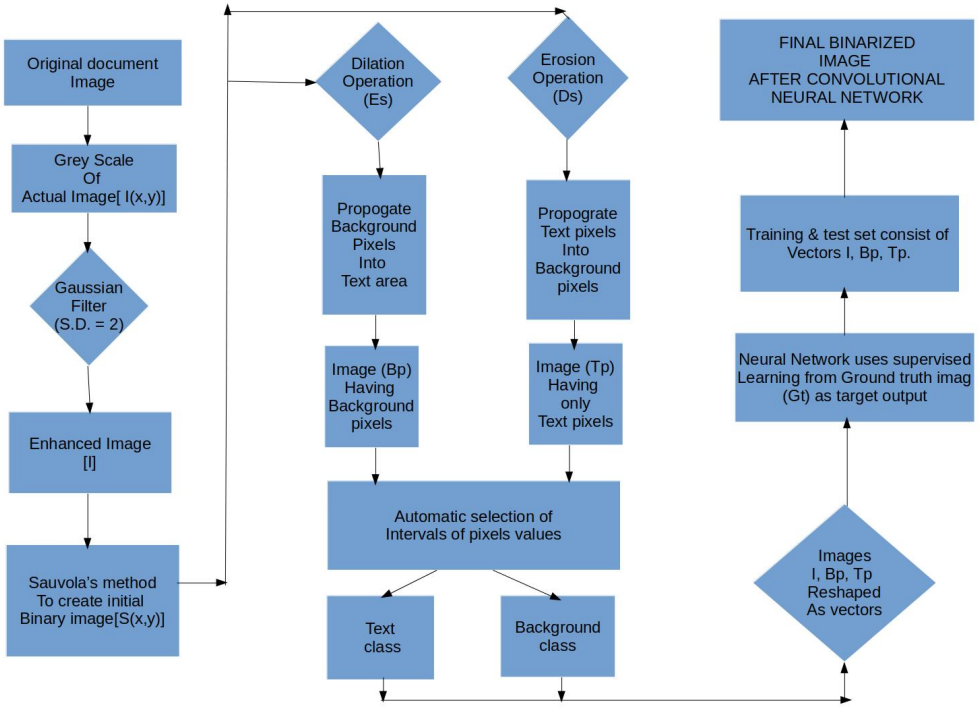

Figure 4.1: Flow chart for the system

4.2 MATHEMATICAL MODEL

Without a solid understanding of deep learning, we can only have a set of empirical rules and intuitions, which is not sufficient to advance the scientific knowledge profoundly. There has been a large amount of efforts devoted to the understanding of CNNs from various angles. In the CNN training, weights are first initialized and then adjusted by back-propagation to minimize a cost function. Each convolutional layer is specified by its filter weights which are determined in the training stage by an iterative update process. That is, they are first initialized and then adjusted by backpropagation to minimize a cost function. All weights are then fixed in the testing stage. These weights play the role of system memory. In this work, we adopt a different name for filter weights to emphasize their role in the testing stage. We call them anchor vectors since they serve as reference signals (or visual patterns) for each input patch of test images. It is well-known that signal convolution can also be viewed as signal correlation or projection. For an input image patch, we compute its

correlation with each anchor vector to measure their similarity. Clearly, the projection onto a set of anchor vectors offers a spectral decomposition of an input.

We define two anchor matrices:

A = [a1, , ak , aK],

B = [b1, , bl , bL]

Clearly, A Rnk and B Rkl . For the correlation part, let y = AT x and z = BT y. Then, we have z = B T AT x = C T x, C AB. Mathematically, we can decompose final CNN output:

x = (N n=1) x(n)e(n)

4.3 DATA FLOW DIAGRAM

4.4 UML DIAGRAM

The following diagrams show the basic working of the neural network of any deep learning project.

The second diagram represent the flow of the fully loaded neural network.

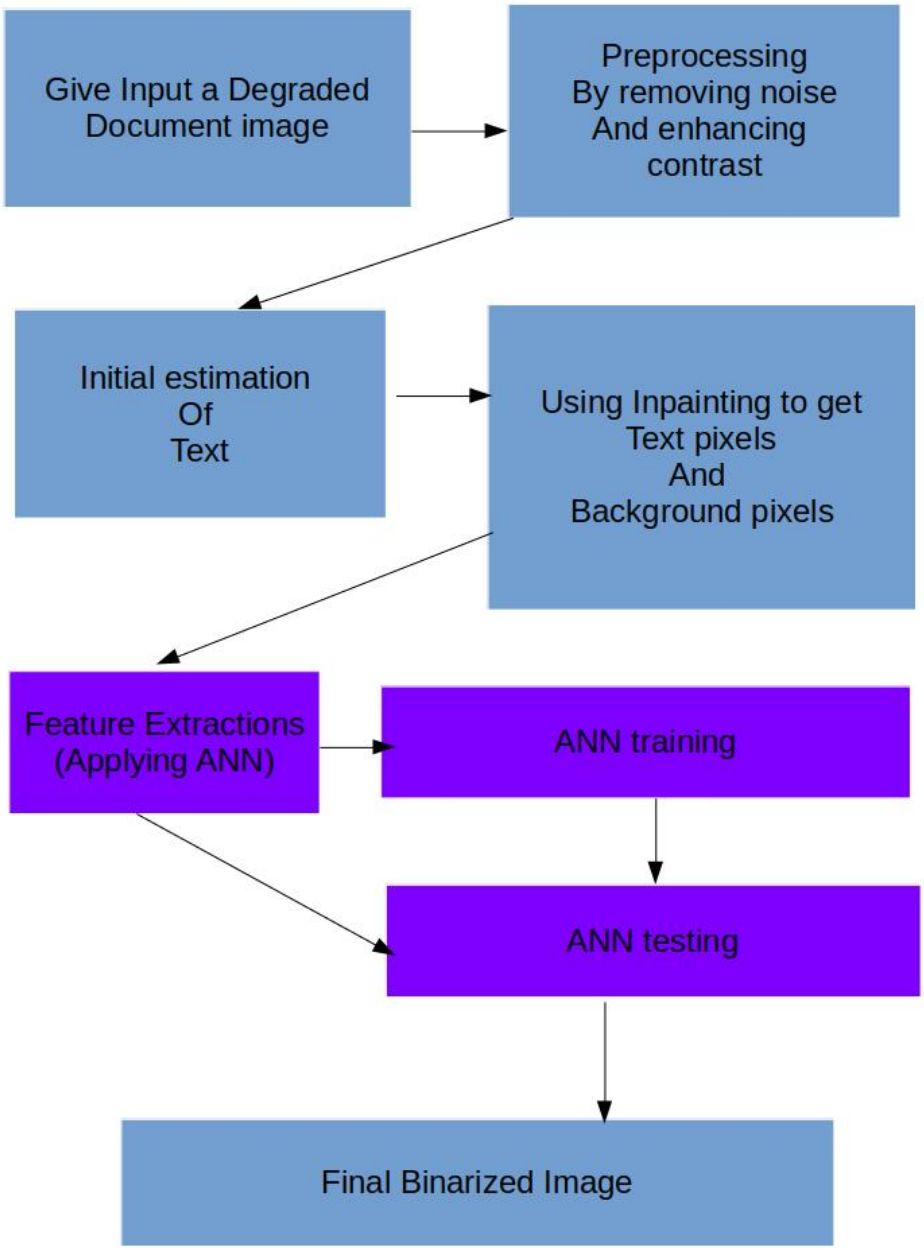

Figure 4.2: Flow chart for the system

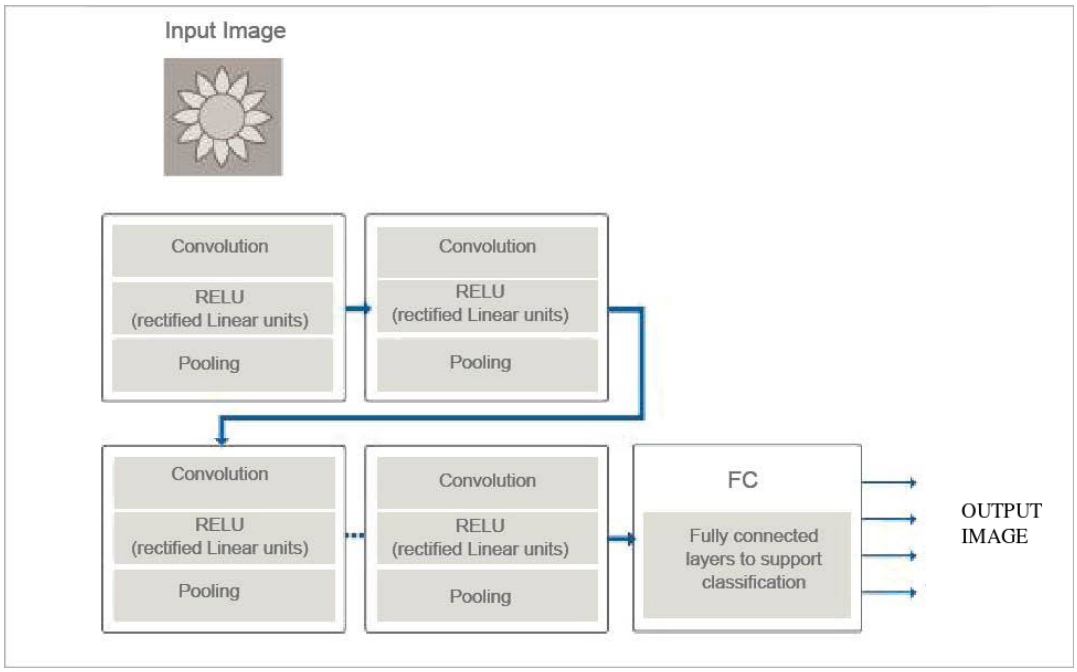

Figure 4.3: Basic Neural Network

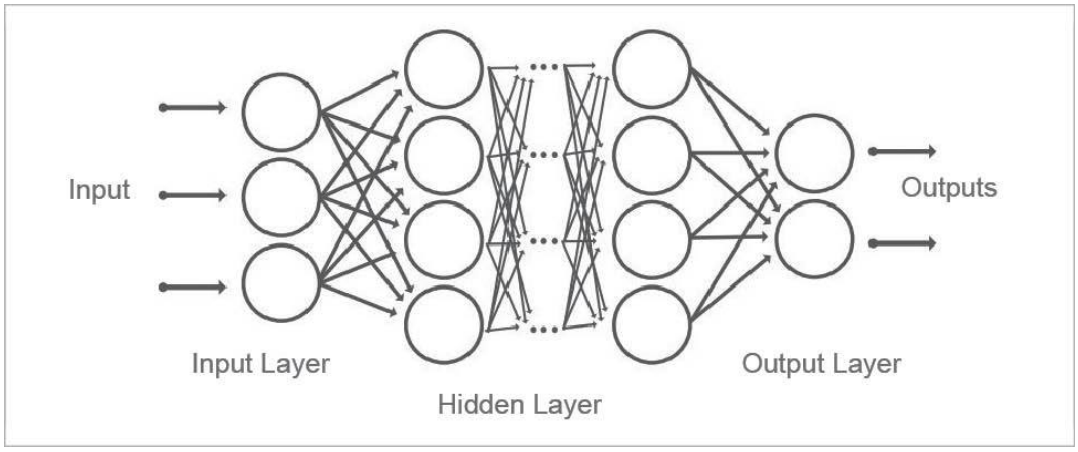

Figure 4.4: Fully loaded Neural Network

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATE

Our project is totally based on the software development which is an application fully developed with all the Open Source tools on the go.

5.1.1 Reconciled Estimate

We are using Keras with Tensorflow in its backend to solve the all deep learning problems. We have used all Open source tools such as Scikit-learn, scikit-image, OpenCV2 and Keras.
For GUI we have deployed Tkinter and Gtk3+. Developemnt time = Effort/Number of days.

5.1.2 Project Resources

Resources used for development:

- Python Platform
- Python GUI framework:
  - Tkinter
  - Glade
- Deep Learning Tool:
  - Keras
  - Scikit-learn
- Image processing tools:
  - OpenCV2
  - Pillow (Python module)
  - Scikit-Image

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

For risk identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories are mentioned.Please refer table 5.1 for all the risks. You can refer following brisk identification questionnaire.

- Have top software and customer managers formally committed to support the project?
- Are end users enthusiastically committed to the project and the system to be built?
- Are requirements fully understood by the software engineering team and its customers?
- Do end users have realistic expectations?
- Does the software engineering team have the right mix of skills?
- Is the number of people in the project team adequate to do the job?
- Do all the user constituencies agree the importance of project and also on the requirements for the system to be build?

5.2.2 Risk Analysis

5.2.3 Overview of Risk Mitigation, Monitoring, Management

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

Major Tasks in project stages are:

- Selection of Project Area
- Requirement Analysis(Base Paper)

Risk description	Probability	Schedule	Impact Quality	Overall
Internet connection not available	Low	Medium	High	High
False review	Low	Low	High	High
Incorrect input	Low	Low	High	High
Adaptability	Medium	Low	Medium	Medium
Efficiency	Low	Low	Low	Low

Figure 5.1: Risk Analysis

Risk ID	1
Risk Description	Internet Connection Not Available
Category	Software Requirement
Probability	Low
Impact	High
Response	Mitigation
Risk Status	Occured

Risk ID	2
Risk Description	False Review
Category	Software Scheduling Risk
Probability	Low
Impact	High
Response	Mitigate
Risk Status	Identified

Risk ID	3
Risk Description	Adaptability
Category	Development Environment
Probability	Medium
Impact	Medium
Response	Management
Risk Status	Can Occur

- Project Specification(Paper Work)
- Algorithm study and Design
- Coding and Implementation(Module Development)
- Project Review
- Testing

Risk ID	4
Risk Description	Efficiency
Category	Functionality
Probability	Low
Impact	Low
Response	Mitigate
Risk Status	Can Occur

Risk ID	5
Risk Description	Incorrect Input
Category	Software Scheduling Risk
Probability	Low
Impact	Low
Response	Mitigate
Risk Status	Identified

- Project Report and Project Delivery

5.3.2 Task Network

5.3.3 Timeline Chart

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Task	Name
Paper Selection	Sayali Mahajan, Kranti Pawar, Sonali Wagh
Literature Survey	Sayali Mahajan, Himanshu Shekhar, Ankush Jadhav
Algorithm Study	Himanshu Shekhar, Ankush Jadhav, Sonali Wagh
Designing	Sonali Wagh, Kranti Pawar, Himanshu Shekhar
Developing	Himanshu Shekhar
Testing	Sonali Wagh, Ankush Jadhav
Deployment	Sayali, Sonali, Kranti, Ankush, Himanshu

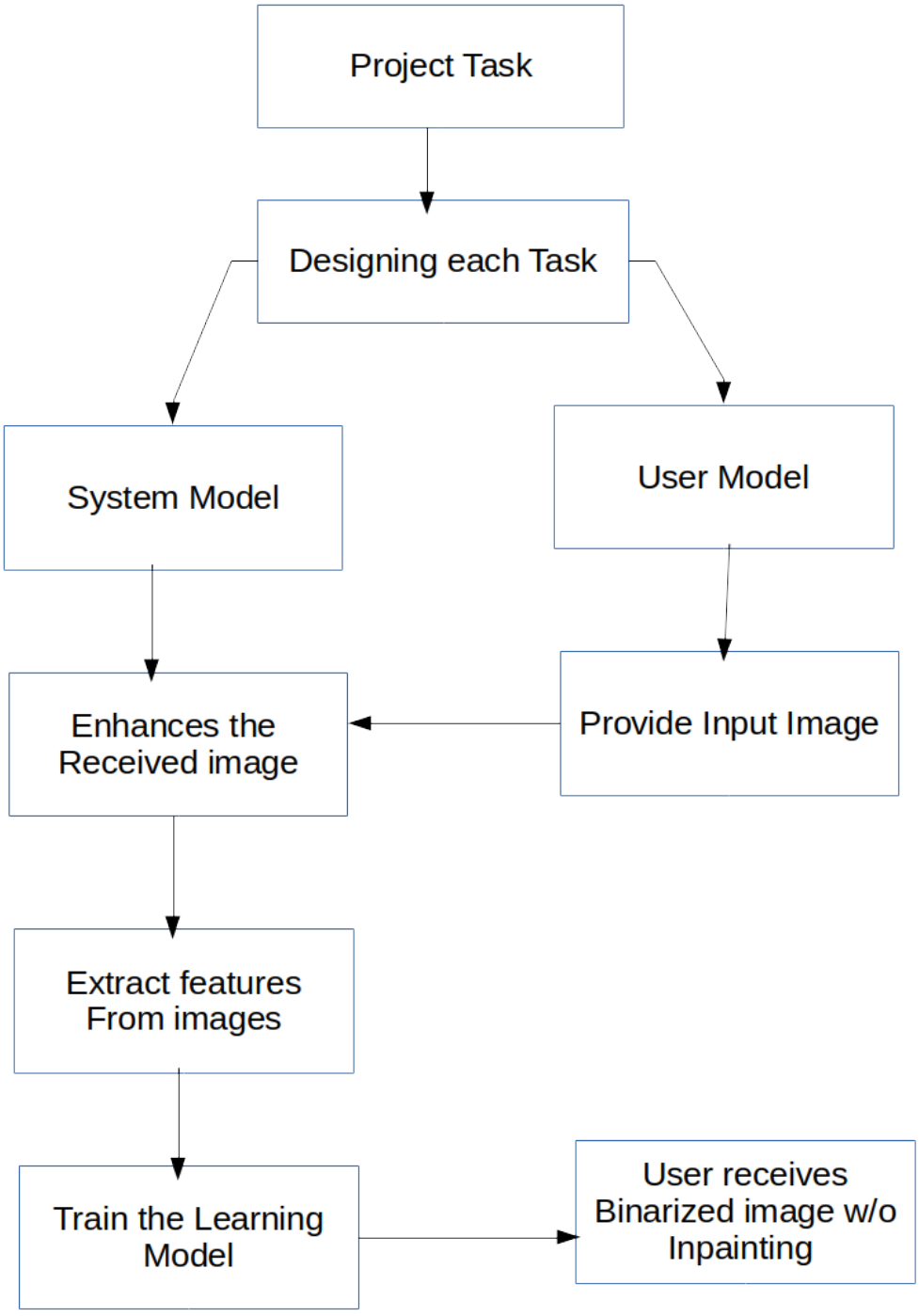

Figure 5.2: Task Network

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

This project consists of total five modules. First module works mainly on the input of image to the system and converting that image to Gray-scale and applying Gaussian Filter to produce enhanced image(I). Second module takes enhanced image(I) as input to the respective algorithm we want to apply, which produces a resulting image [S(x,y)]. Third module focuses on Morphology technique i.e. applying dilation operation and erosion operation simultaneously. Dilation operation propagate text pixels into background pixels giving us text pixels(Tp) where as erosion operation propagate the background pixels into text area giving us background pixels(Bp). Fourth module is created so that the model can automatically make selection of intervals of pixel values resulting in Text class and Background class respectively. Then these pixels are reshaped vectors which are used for features selection and feature validation of the images. Final module will do the task of splitting the data-set into training and testing data-sets. Our model is trained using training data-set and learns from available features. This module also display the output image of the system along with necessary output which help us in understanding the model such as: Ground Truth, Accuracy, etc.

6.2 TOOLS AND TECHNOLOGY USED

Various tools are used while developing this project. Our whole project is developed in Python3.5+ and various python modules like Scikit-learn, Scikit-image, Numpy, Scipy, Matplotlib, OpenCV3.
Graphical User interface is also developed in python GUI framework Tkinter and Glade.
Deep Learning tools such as Keras having Tensorflow in backend.

6.3 ALGORITHM DETAILS

We have give option to the user that which algorithm they want to perform on the input image.

```
max,s,t = 0;
for ss: 0 to L-1 do
    for tt: 0 to L-1 do
        evaluate tr(S_b);
        if tr(S_b) > max
            max = tr(S,b);
            s = ss;
            t = tt;
        end if
    end for
end for
return s,t;
```

Figure 6.1: Otsu's Method

6.3.1 Otsu's Algorithm

Otsu's method is used to automatically perform clustering-based image thresholding, or, the reduction of a graylevel image to a binary image.The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal.

6.3.2 Sauvola's Method

Sauvolas method takes a grayscale image as input. Since most of document images are color images, converting color to grayscale images is required. For this purpose, we choose to use the classical luminance formula, based on the eye perception:

Luma = 0.299 R + 0.587 G + 0.114 B

From the grayscale image, Sauvola proposed to compute a threshold at each pixel using:

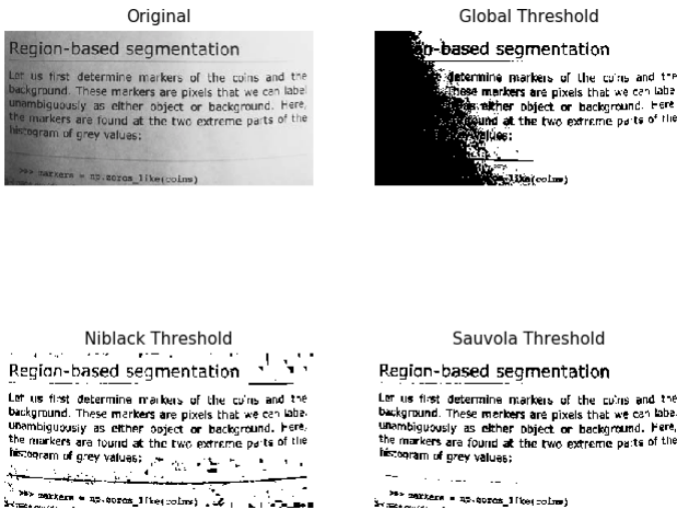

Figure 6.2: Comparision between different algorithms

T = m [1 + k (s/R 1)]

.

6.3.3 Niblack's Algorithm

The concept of Niblack, algorithm is to build a threshold surface, based on the local mean, m. and local standard deviation,s, computed in a small neighborhood of each pixel in the form of:

T = m + k * s

where k is negative constant.

6.3.4 How Convolutional Neural Network work?

Convolutional Neural Nets are usually abbreviated either CNNs or ConvNets. They are a specific type of neural network that has very particular differences compared to MLPs. Basically, you can think of CNNs as working similarly to the receptive fields of photoreceptors in the human eye. Receptive fields in our eyes are small connected areas on the retina where groups of many photo-receptors stimulate much fewer ganglion cells. Thus, each ganglion cell can be stimulated by a large number of receptors, so that a complex input is condensed into a compressed output before it is further processed in the brain.

We need a numerical representation of our image because just like any other machine learning model or neural net, CNNs need data in form of numbers in order to learn! With images, these numbers are pixel values; when we have a grey-scale image, these values represent a range of greyness from 0 (black) to 255 (white).
Pooling layers are used to reduce the size of images in a CNN and to compress the information down to a smaller scale. Pooling is applied to every feature map and helps to extract broader and more general patterns that are more robust to small changes in the input. Common CNN architectures combine one or two convolutional layers with one pooling layer in one block. Several of such blocks are then put in a row to form the core of a basic CNN.Pooling layers also work with sliding windows; they can but dont have to have the same dimension as the sliding window from the convolutional layer. Also, sliding windows for pooling normally dont overlap and every pixel is only considered once. There are several options for how to pool:

- max pooling will keep only the biggest value of each window.
- average pooling will build the average from each window.
- sum pooling will build the sum of each window.

After our desired number of convolution + pooling blocks, there will usually be a few dense (or fully connected) layers before the final dense layer that calculates the output. These dense layers are nothing else than a simple MLP that learns the classification or regression task, while you can think of the preceding convolutions as the means to extract the relevant features for this simple MLP.

CHAPTER 7

SOFTWARE TESTING

7.1 TYPES OF TESTING

- Unit Testing

  Each module is tested separately. Components of unit testing are original single component. Firstly, unit testing of preferences is done. It is checked if all the requirement are covered as per preferences given by user. Each module is tested separately. Components for unit testing are original dataset. The resizing of the all the dataset images are done before testing and implementing.

- Integration Testing

  In integration testing, the modules which are unit tested are combined and testing is performed to see if the correct information is passed between the modules as per the algorithms.

- GUI Testing

  We have tested the user interface to check the functionality of all the buttons and control panel components of leap motion controller.

CHAPTER 8

RESULTS

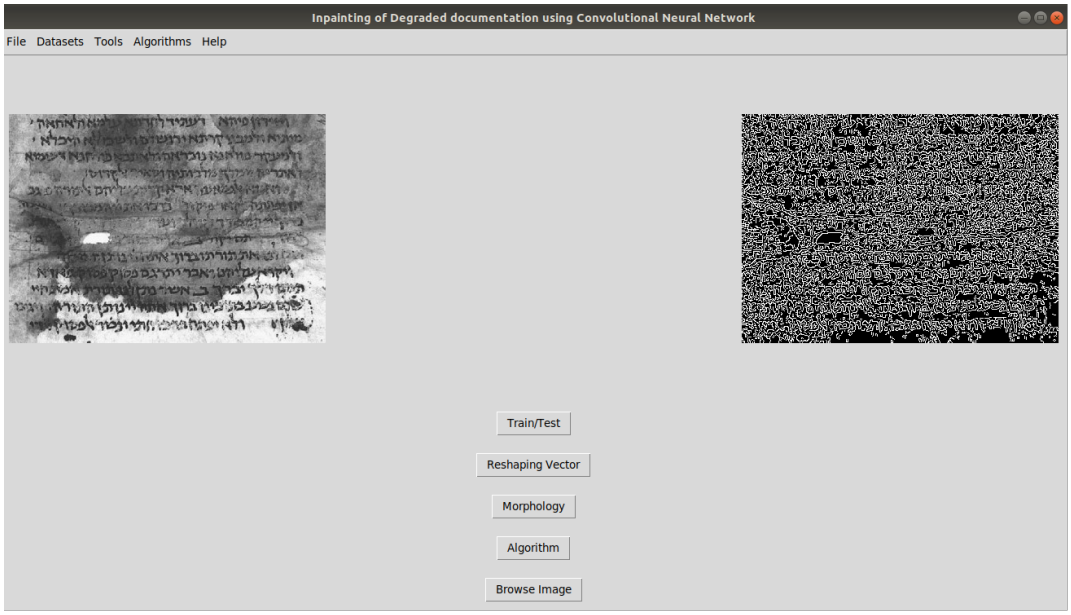

Figure 8.1: First screen short of the system.

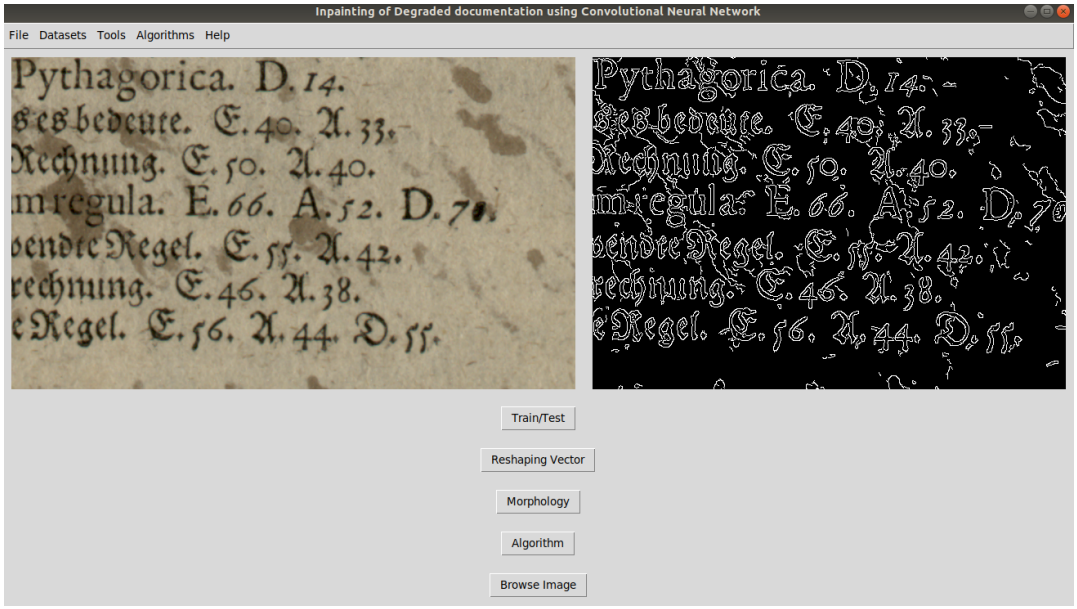

Figure 8.2: System with other images.

8.1 OUTCOMES

The outcome of the system should be the binarized image after applying the inpainting techique. The binarized image must consitst of two pixels namely black(0) and white(1).

8.2 SCREEN SHOTS

CHAPTER 9

CONCLUSIONS

9.1 CONCLUSION

This report will show how our system is able to learn from the input degraded image datasets. We used image pre-processing to enhance luminosity and threshold images. We then used Keras to extract features from the input images and these extracted features are used further to train the Convolutional Nerual model.
As perspectives, we would like to understand the backend of the image processing using deep learning, especially for the degraded document images. Somewhere we also tried to combine some learning based methods so that the inpainting technique can be used to reconstruct textual images.

9.2 FUTURE WORK

- Image Inpainting for Irregular Holes Using Partial Convolutions[**?**] Partial Convolutions

9.3 APPLICATIONS

Inpainting methods based on neural networks:

- Deep Image Prior[1]
- High-Resolution Image Inpainting[3]
- Generative Image Inpainting With Contextual Attention[2]
- Content-Aware Fill in Adobe Photoshop CS5.[**?**]

ANNEXURE A

- Naouel Ouafek, Mohamed-Khireddine Kholladi,"A Binarization Method for Degraded Document Image using Artificial Neural Network and Interpolation Inpainting". IEEE 2018.
- M. Bertalmio and G. Sapiro and V.Caselles and C. Ballester, Image Inpainting, Proceding of SIGGRAPH, 417424, 2000
- N. Otsu, A threshold selection method from gray level histogram,IEEE Trans. Syst., Man, Cybern,vol. 19 no. 1,6266,Jan. 1979
- W. Niblack,An Introduction to Digital Image Processing,115-116,1986
- Sauvola and M. Pietikainen, Adaptive document image binarization,Pattern Recognition, vol. 33, no. 2, 225236, 2000
- Gatos, B., K. Ntirogiannis, and I. Pratikakis. 2009a. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009) . In ICDAR09. p. 13751382.
- B. Gatos and I. Pratikakis and S. Perantonis, Adaptive degraded document image binarization, In Pattern Recognit, 39, 3173278,2006.
- J.Canny, A computational approach to edge detection, In IEEE Trans.PatternAnal. Mach. Intell.8(6), 679698, 1986
- J. Sauvola, M. Pietika K inen, Page segmentation and "classification using fast feature extraction and connectivity analysis, International Conference on Document Analysis and Recognition", ICDAR '95, Montreal, Canada, 1995,pp. 11271131.
- Jyoti Rani and Davinder Parkash, "Degraded Document Image Binarization Techniques", IJCET-Dec-2015,E-ISSN 2277 4106, P-ISSN 2347 5161.

ANNEXURE B

PLAGIARISM REPORT

Sr.No.	Chapter	Result
1.	Abstract	91%
2.	Synopsis	95%
3.	Introduction	80%
4.	Problem Definition & Scope	100%
5.	Project Plan	100%
6.	Software Requirement & Specification	100%
7.	Project Implementation	86.4%
8.	Software Testing	80%
9.	Conclusion & Future Scope	89.3%

ANNEXURE C

REFERENCES

- Naouel Ouafek, Mohamed-Khireddine Kholladi,"A Binarization Method for Degraded Document Image using Artificial Neural Network and Interpolation Inpainting". IEEE 2018.
- M. Bertalmio and G. Sapiro and V.Caselles and C. Ballester, Image Inpainting, Proceding of SIGGRAPH, 417424, 2000
- N. Otsu, A threshold selection method from gray level histogram,IEEE Trans. Syst., Man, Cybern,vol. 19 no. 1,6266,Jan. 1979
- W. Niblack,An Introduction to Digital Image Processing,115-116,1986
- Sauvola and M. Pietikainen, Adaptive document image binarization,Pattern Recognition, vol. 33, no. 2, 225236, 2000
- Gatos, B., K. Ntirogiannis, and I. Pratikakis. 2009a. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009) . In ICDAR09. p. 13751382.
- B. Gatos and I. Pratikakis and S. Perantonis, Adaptive degraded document image binarization, In Pattern Recognit, 39, 3173278,2006.
- J.Canny, A computational approach to edge detection, In IEEE Trans.PatternAnal. Mach. Intell.8(6), 679698, 1986
- J. Sauvola, M. Pietika K inen, Page segmentation and "classification using fast feature extraction and connectivity analysis, International Conference on Document Analysis and Recognition", ICDAR '95, Montreal, Canada, 1995,pp. 11271131.
- Jyoti Rani and Davinder Parkash, "Degraded Document Image Binarization Techniques", IJCET-Dec-2015,E-ISSN 2277 4106, P-ISSN 2347 5161.

REFERENCES

[1] Victor Lempitsky Dmitry Ulyanov, Andrea Vedaldi. *Deep Image Prior*. Submitted on 29 Nov 2017 (v1), last revised 5 Apr 2018 (this version, v3).

[2] Yu et al. *Generative Image Inpainting With Contextual Attention*. 2018.

[3] Simo-Serra Iizuka and Ishikawa. *Globally and Locally Consistent Image Completion*. 2017.