CSE 326: Software Engineering

Final Project Proposal & Plan

Authors:

Cole Johnson - *cole.johnson@student.nmt.edu*
Colin Grandjean - *colin.grandjean@student.nmt.edu*
John Runyon - *john.runyon@student.nmt.edu*
Lauren Giles - *lauren.giles@student.nmt.edu*

New Mexico Institute of Mining and Technology
Socorro, NM 87801, USA

Date: February 13, 2025

Contents

1 Introduction

1.1 Project Overview and Statement of Proposal

OCR (Optical Character Recognition) is a technique that a program can use to parse individual or stream images into a matched set of some written alphabet; often just a set of alphanumeric characters. OCR is used in banking, note taking applications, and many other services used on a daily basis. One of the most common implementations of OCR is the use of neural networks, often through supervised learning methods. We propose to create a simple neural network that will be trained and tested to classify a single image into an alphanumeric character using one such supervised learning technique. In the process of developing the neural network, we'll develop a host of tools to test and view the networks created and some performance metrics to gauge effectiveness.

There are a few motivations for the project. First, the development of this tool can help people learn more about neural networks and how they work using our graphical user interface. Second, OCR has a manifold of applications, as discussed previously, so the tool we create could be used for a variety of applications – such as written to digital text translation software.

1.2 Project Scope and Objectives

The initial scope of the project, as outlined in the overview, will be to create a simple Optical Character Recognition (OCR) system using a neural network and employ supervised learning techniques to test and train our model. Our main objectives for the project include:

(a.) Build the components of neural network using object-oriented design principles and programming language (like Java).

(b.) A training and test environment for our neural network.

(c.) A graphical user interface (GUI) that the user can draw characters for the model to classify them.

(d.) A graphical user interface (GUI) to view the created neural networks to visually see how each model works.

2 Risk Management Strategy

2.1 Risk Table

Category Values

- **BU** - Business Impact Risk
- **CU** - Customer Risk
- **DE** - Development Environment Risk
- **PR** - Process Risk
- **PS** - Product Size Risk
- **ST** - Risk Associated with Staff Size and Experience
- **TE** - Technology Risk

Impact Values:

- **4** - catastrophic
- **3** - critical
- **2** - marginal
- **1** - negligible

Table 1: Risk Table

Risks	Category	Probability	Impact	RMMM
Project scope may increase due to feature additions	**PS**	Possible	2	**Management** - Define strict requirements at the beginning and have periodic reviews to control feature creep.
Model may not generalize well to unseen characters	**PR**	Likely	3	**Management** - Incorporate diverse training data, test edge cases, and implement validation methods. **Monitoring** - use performance metrics to validate training data
Members use a variety of software and hardware for development	**DE**	Very Likely	1	**Mitigation** - Use a virtual machine to help simplify the process of developing the software and make diagnosing issues easier by sticking to one platform.
No robust method for testing whether our model works	**PR**	Very Likely	2	**Mitigation** - Introduce unit tests for individual components of neural network.

Continued on the next page

Continued from previous page				
Risks	**Category**	**Probability**	**Impact**	**RMMM**
Some group members are inexperienced with neural networks	**ST**	Likely	3	**Management** - Hold meetings to review neural network knowledge and carefully plan the learning strategy, including which technologies to involve.
Technology stack may not support required performance	**TE**	Possible	3	**Monitoring** - Evaluate and select technologies that are proven for neural network applications.

2.2 Discussion of Risks to Be Managed

The risk to be managed for this project will primarily involve managing the technical complexity, rather than dealing with public-facing risks like product size or customer relations. The risks for this project require mitigation, management, and monitoring measures that attempt to alleviate primarily technical or staff risks. For example, one mitigation effort requires unit testing to help ensure the behavior of each component of the neural network is functioning as expected.

2.3 Risk Mitigation, Monitoring, and Management Plan

2.3.1 Risk Mitigation

Mitigation efforts focus on reducing technology risks. A few mitigation efforts involved in the project are discussed in greater detail here.

1. **No Robust Testing Method to Evaluate the Whole Neural Network - Process Risk:**

   It's very difficult to evaluate how a neural network is flawed, since it's connections are often elaborate and difficult to understand (the black box effect). Instead, to help mitigate this risk we propose to introduce unit testing inside the components, and test smaller version of the models we will ultimately create.

2. **Inexperience in Neural Networks - Staff Risk:**

   If this risk becomes a problem, our mitigation efforts involve holding meetings to review relevant information about neural networks. Writing unit tests should also

help, since they help the writer understand how each component is supposed to function

2.3.2 Risk Monitoring

Risk monitoring involves gauging understanding of each component of the project, ensure that each part is properly understood and that tests are written properly to ensure each component is working as intended and understood by members of the project.

2.3.3 Risk Management (Contingency Plans)

Contingency plans to reduce risk involve creating and utilizing many different neural network performance metrics, like hamming distances and F1 scores, the ensure the accuracy of our model and the soundness of our training process. If we encounter issues implementing the supervised learning process, we can try different objective functions and performance metric that are easier to evaluate and control – allowing for some degree of freedom in implementation and the development of contingencies as required.

1. **Model may not generalize well to unseen characters - Process Risk:**

   The performance and effectiveness of a neural network hinges on the quality and quantity of a dataset. In order to improve results, we should constantly be evaluating our dataset and implemented steps for processing that data.

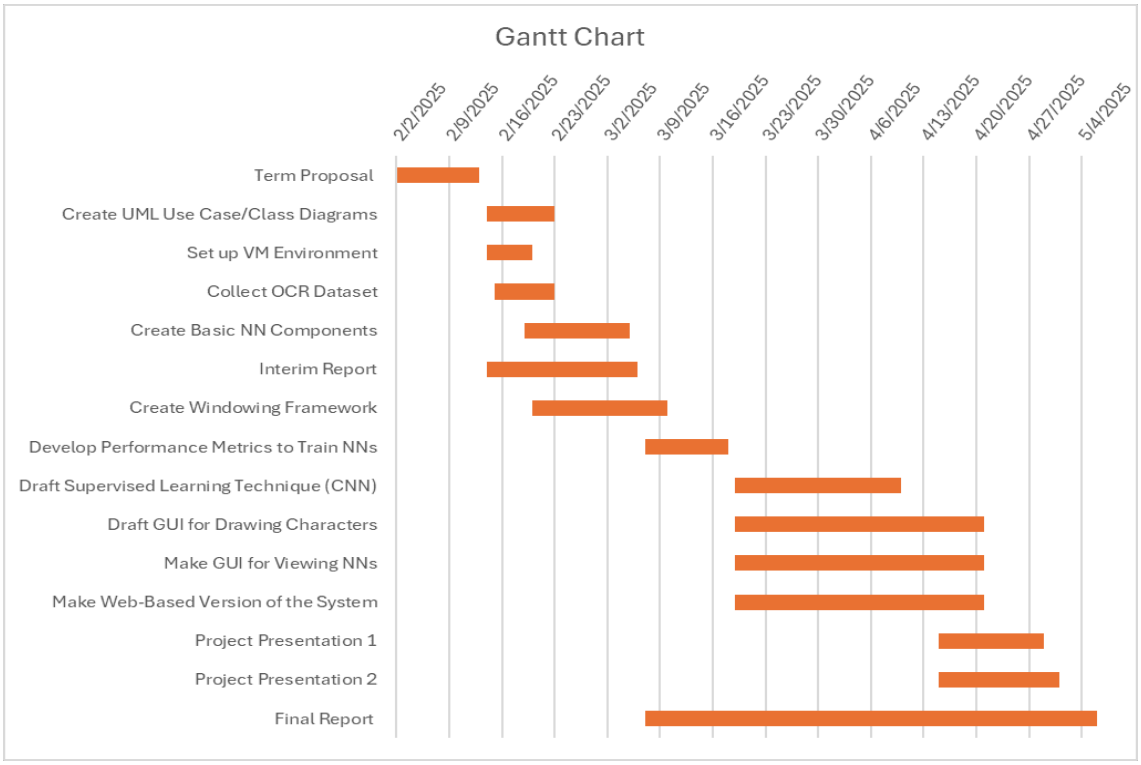
3 Schedule

3.1 Task List

1. Term Proposal
2. Create UML Use Case/Class Diagrams
3. Set up VM Environment
4. Collect OCR Dataset
5. Create Basic NN Components
6. Interim Report
7. Create Windowing Framework

8. Develop Performance Metrics to Train NNs
9. Draft Supervised Learning Technique (CNN)
10. Draft GUI for Drawing Characters
11. Make GUI for Viewing NNs
12. Make Web-Based Version of the System
13. Project Presentation 1
14. Project Presentation 2
15. Final Report

3.2 Timeline Chart



Gantt Chart of Main Tasks for the Project

3.3 Resource Table

Table 2: Resource Table

Task	People	Hardware and Software	Special
Term Proposal	Group	Latex and Excel	None
Create UML Use Case/Class Diagrams	Group	*Dia* and *draw.io*	None
Set Up VM Environment	John Runyon	Computer and VM	None
Collect OCR Dataset	Colin and John	Computer and VM	None
Create Basic NN Components	Cole Johnson	*openJDK22* and *Java Language*	None
Interim Report	Group	*LaTex*	None
Create Windowing Framework	Lauren	*openJDK22* and *Java Language*	None
Develop Performance Metrics to Train NNs	Cole and John	*openJDK22* and *Java Language*	None
Draft Supervised Learning Technique (CNN)	Group	Computer and VM	None
Draft GUI for Drawing Characters	Lauren	*openJDK22* and *Java Language*	None
Make GUI for Viewing NNs	Cole and Lauren	*openJDK22* and *Java Language*	None
Make Web-Based Version of the System	John	*nginx*, *JSweet*, and *React*	None

Continued on the next page

Continued from previous page			
Task	**People**	**Hardware and Software**	**Special**
Project Presentation	Group	*Google Slides* and *LaTex*	None
Final Report	Group	Latex	None

4 Project Resources

4.1 People

1. Colin Grandjean
2. Lauren Giles
3. Cole Johnson (Team Leader)
4. John Runyon

4.2 Hardware and Software Resources

We plan to create our project using Java, version 22. To help automate our build and deployment processes, we are using Gradle. In order to facilitate version control, we use GitHub with Jira to keep track of our development process and task management. Each team member uses different code editors, depending on their preferences. We will be using primarily Latex to write our reports and keep track of documentation.

4.3 Special Resources

The only possible special resource we may need for the project is a Virtual Machine to manage the development environment. Later, we will use special software to try and port our java application to web, which may need special considerations.

5 Appendices

5.1 Tasks Completed by Group Members

Cole: Drafted section 1 and section 2 of the proposal, and wrote some entires of the risk table. Organized meetings and maintained a git repository for work on the

final project. Brainstormed ideas and organized a task list along with all the other members of the project.

Colin: Created task list, Gantt Chart, and Resource table. Transferred all of the information from the rough draft of our task list, which included tasks, dates, and people who are working on the tasks, into the project proposal. Completed all of section 3 except for the Hardware and Software section of the resource table.

John: Assisted in overall document formatting, including extended tables, alphabetical organization, and the risk table. Reviewed and improved grammar throughout the document. Contributed to meeting planning and organization to enhance workflow, streamline ideas and scope, and suggested process management strategies and software for project completion.

Lauren: Created the Jira project for tracking issues and overall development process using an Agile workflow as well as configuring the project to prepare for the start of the development process. Added information for the Project requirements section and aided in some of the planning of needed tasks during team meetings.