Final sprint

Bartholomew Binkleburg

Sprint Review Report

Sprint 8

5/5/25

<Mitch, Astrid, Andrew>

Software Engineer

Prepared for

<SP25-CSCI4099-01>

<Spring 2025 >

<La Roche University>

# Revision History

| **Date** | **Description** | **Author** | **Comments** |
| --- | --- | --- | --- |
| <3/3/2025> | <Version 1> | <Andrew Booth> | <template> |
| 5/5/2025 | final version | Astrid Lutz |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

| **Signature** | **Printed Name** | **Title** | **Date** |
| --- | --- | --- | --- |
|  | Andrew Booth | Software Eng. | 5/5/25 |
|  | Mitch Wright | Software Eng. | 5/5/25 |
|  | Astrid Lutz | Software Eng. | 5/5/25 |

**Table of Contents**

**Revision History ii**

**Document Approval ii**

**1. Introduction 1**

**2. Specific Goals 1**

**3. AnalTICS 2**

**4. CONCLUSION 3**

# 1. Introduction

*Objectives were to connect php, database, and flask operations seamlessly. As well as to address errors in the model not training on certain architectures as well as stagnating in a valley of no box predictions*

# 2. Specific Goals

*This section should include specific stories planned for the Sprint:*

***2.1 Story Name & Number***

*Story Name: Front end back end sync.*

*Number: #008*

***2.1.1 Story Description:***

*There is a page that the user can access in order to view their account info and previous searches*

***2.1.2 Story Acceptance Criterion***

* *New page is created*
* *Search bar sends to backend*
* *Returns class images filtered by color.*
* *There is an area for previous image searches*

***2.1.3 Story Dependencies***

*This story is dependent on the previous front end implementation/connection with the database*

***2.1.4 Story Challenges***

*There were no challenges.*

***2.1.5 Story Assigned to***

*This story was assigned to Mitch and Andrew*

***2.1.6 Story Points***

*8 points*

***2.1.7 Status: Completed or not***

*This Story was completed.*

***2.2 Story Name & Number***

*Story Name: Correcting the Mask RCNN*

*Story Number: #09*

***2.2.1 Story Description:***

*Attempting to address failure to train, sometimes even to init.*

***2.2.2 Story Acceptance Criterion***

* *Masks created*
* *Boxes created*
* *Classes accurate*

***2.2.3 Story Dependencies***

*This story was dependent on previous versions of the model.*

***2.2.4 Story Challenges***

*Everything either crashed the session, caused the gradients to blow up, or caused the model to not return boxes, possibly due to loss favoring neutral IOU. Ended up creating a naive search approach by creating a pkl file of all image class contents and mode color based off of the masks and selecting the mode color within the bounding box created from them.*

***2.2.5 Story Assigned to***

*This story was assigned Astrid.*

***2.2.6 Story Points***

*32 points*

***2.2.7 Status: Completed or not***

*This story was not completed*

## 

# 3. Conclusion

Sprint 8 was a success on the part of the website functionality. Sprint 8 focused on completing the Flask backend and integrating image and attribute-based search features. The /upload and /search routes were finalized, enabling users to retrieve matches through both AI simulation and real class-color filtering. These additions mark the completion of core functionality and significantly enhance the platform’s search capabilities.

On the part of the Mask RCNN, a test in patience and a learning opportunity. It provided an opportunity to learn more about hardware limits of models, how to diagnose problems that do not throw errors, the architecture of models like fastRCNN and ROI heads, how to work with learning rates, a bit about collinearity and L2 normalization, preprocessing procedures, freezing and unfreezing layers of a pretrained model, and how models will find naive solutions to minimizing loss in a local valley. A naive approach to the search function eventually had to be used by creating a pkl file of all image class contents and mode color based off of the masks and selecting the mode color within the bounding box created from them.