ENPH 455 Final Report

Streamlining Hydride Annotation for Canadian Nuclear Labs

by

J. Everitt

A thesis submitted to the
Department of Physics, Engineering Physics and Astronomy
in conformity with the requirements for
the degree of Bachelor of Applied Science

Queen's University
Kingston, Ontario, Canada
April 2023

Executive Summary

Delayed hydride cracking (DHC) is a critical corrosion issue in CANDU reactors. The client, Canadian Nuclear Labs (CNL), utilizes transmission electron microscopy (TEM) imaging to study hydride properties. Since manual annotation of the TEM images is a tedious, time-consuming process, CNL partnered with Queen's University to automate the procedure. This led to the development of a neural network for image segmentation using the state of the art Mask R-CNN architecture. The objective of this thesis was creating a tool that allows CNL microscopists to use the neural network without the help of an AI engineer and seamlessly revise any faulty annotations.

The tool must display original images and Mask R-CNN generated annotations to the microscopist, provide editing capabilities for revisions, and store revised annotations in an appropriate format. It should reduce annotation turnaround time from 3 days to 1.5 days. It should also be simple for an end user with minimal training to use. Key constraints were storage and speed. The tool should occupy minimal storage for ease of sharing and display images at a reasonable speed, taking at most one minute to run an image through the Mask R-CNN model.

Background research was conducted into TEM imaging, the importance of studying DHC, creating and sharing virtual machines (VMs), and the Mask R-CNN architecture. Various test jobs were also submitted to the Digital Research Alliance of Canada compute clusters to gain familiarity with how the Mask R-CNN model is trained.

Tkinter was used to build an image annotation tool that allows users to draw polygonal

annotations directly on an image. This was then integrated with the Mask R-CNN model to provide automatic annotation functionality. The product was transferred to a VM, uploaded to the CNL server, and imported by the client into VirtualBox software for use. Secure file transfer protocol (SFTP) was used for all file sharing to mitigate security risks.

The tool was tested by CNL stakeholders and determined to be a success. The design was iterated to incorporate their suggestions for improvement. The final product annotates images at a rate of 40 seconds per image and occupies 15 GB of storage. The GUI has 7 buttons that allow the user to easily navigate through and annotate images. The tool uses entirely free and open source software, however a membership to the Digital Research Alliance of Canada compute clusters was required to train the Mask R-CNN model, valued between \$5,000 and \$15,000 annually.

A new tool feature that integrates manual annotations with Mask R-CNN generated annotations for a single output CSV file will be completed by the end of the semester. This was not identified in the original scope. Additional testing and iteration to improve VM performance will also be conducted. Future work includes extending the tool to the annotation of grain boundaries, which is another property that provides insight into the degradation of CANDU reactors.

Acknowledgments

Thank-you to my supervisor, Professor Laurent Béland, for making this project possible, and to Jun-Tiang Zhang and Yezhou Ni for their support. Thank you to Travis Skippon, Sean Hanlon, and Mark Daymond for their constructive feedback and support with implementation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The following contains background information on the problem and proposed solution.

1.1 Motivation and Problem Definition

Delayed hydride cracking is one of the most critical corrosion issues in the primary heat transport system of CANDU reactors [1]. The hydride is prone to cracking under stress, which has been the source of major replacements in the past. The client, Canadian Nuclear Labs (CNL), utilizes transmission electron microscopy (TEM) imaging to study hydride properties such as size and orientation relative to the alloy's texture. Since manual annotation of the TEM images is a tedious, time-consuming process, CNL has partnered with Queen's to automate the procedure. This led to the development of artificial neural networks for image segmentation.

Queen's Computational Materials Science research group uses the VGG Image Annotation Tool developed by Oxford University to draw polygonal shapes around the hydrides, which serve as the "ground truth" for training the neural networks [2]. Using a set of a 150 manually annotated micrographs, the group developed a hydride detection tool based on the Mask R-CNN architecture, which is a state-of-the-art model for image segmentation [3]. They also are in the process of developing tools based on the Mask R-CNN and U-Net architectures to automatically annotate grain boundaries, which is another property that provides insight into the degradation of the nuclear reactor [4]. Figure 1.1 shows an original TEM image and the hydride annotations generated using Mask R-CNN.

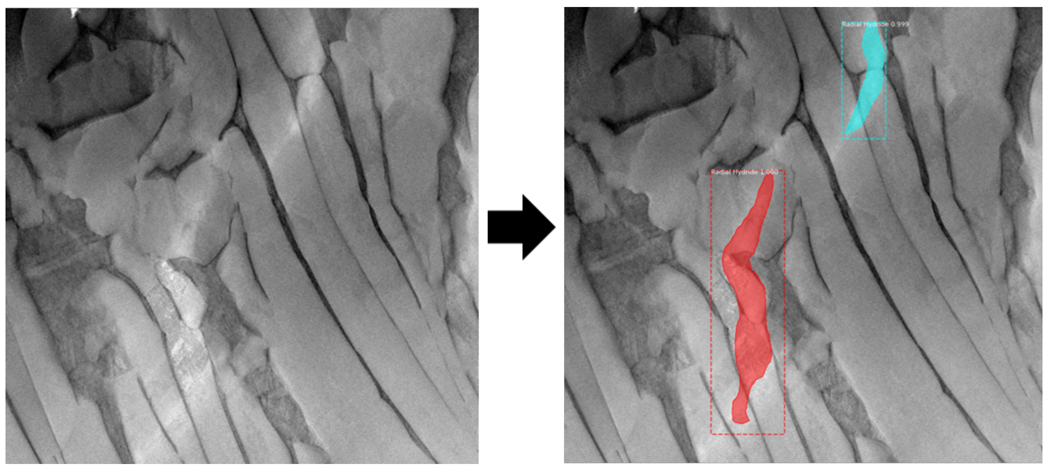

Figure 1.1: TEM image before and after hydride annotation using Mask R-CNN. Labels on annotated image read "Radial Hydride 1.000" (red) and "Radial Hydride 0.999 (blue)," where the values indicate confidence.

The original process for image annotation using the neural network often required several rounds of back-and-forth due to imperfect results. This created motivation to streamline the process and reduce AI engineer and microscopist time required to analyze the hydrides. The original flow and proposed new flow for hydride annotation using Mask R-CNN are shown in Figure 1.2 and Figure 1.3 below.

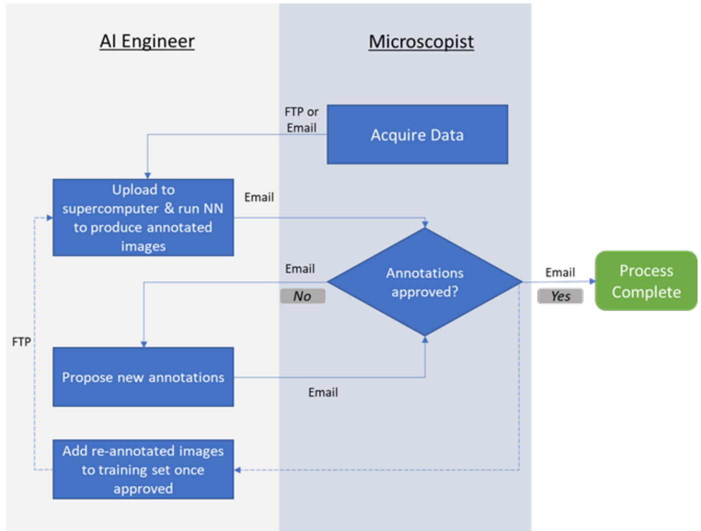

Figure 1.2: Original TEM image annotation process between AI engineer and CNL microscopist.

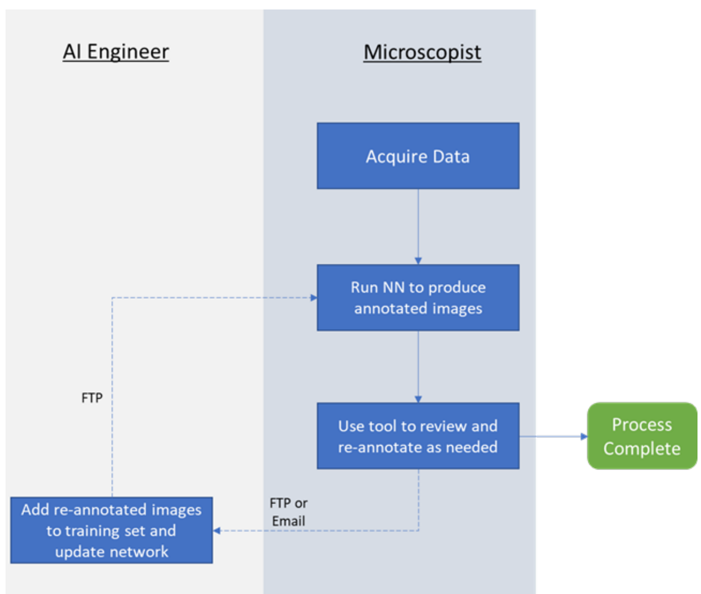

Figure 1.3: Target TEM image annotation processes between AI Engineer and CNL Microscopist.

1.2 Objectives

The focus of this thesis was developing a tool with a simple graphical user interface (GUI) that would allow CNL microscopists to annotate images and revise annotations independently to transition from the current to proposed process flow. As a stretch-goal, this tool could be further applied to the identification of grain boundaries in TEM.

1.3 Functional Requirements and Constraints

The tool must display original annotations to the microscopist, provide editing capabilities for revisions, and store revised annotations in an appropriate format. A user-friendly interface is required to interact with these features. The final product should reduce annotation turnaround time for the end user by 50%, from an estimated 3 days to 1.5 days. This is a conservative estimate as the original process is not robust and requires both the AI engineer and the microscopist to be available, which breaks down if one of the parties takes a sick leave.

The product must be simple for an end user with minimal training to use. The user should not have to download a collection of software libraries in order to run the model, nor should they require an understanding of the neural network architecture to interact with the tool. Re-annotated images must be saved to a new folder such that these can be shared with the Queen's University team to re-train the neural network.

Key constraints are storage requirements and speed. A product that occupies minimal storage will be easier to share with the client and set up. The annotation correction tool must also display images at a reasonable speed, taking 1 minute at most to run an image through the Mask R-CNN model and display annotations. Additionally, the user should be able to fix annotations and navigate through raw images with no visible lag.

1.4 Background Research

Background research was conducted to understand TEM imaging, hydride formation, the machine learning model architecture, and the tools required to build the product.

1.4.1 TEM Imaging

TEM is a microscopy technique that generates high resolution images at the Angstrom scale, allowing for effective study of materials at an atomic level [5]. The microscope emits electrons from a source through a vacuum, which are then focused through an electromagnetic lens into a thin beam. This beam is directed through the material of interest onto a fluorescent screen. A schematic of a transmission electron microscope is shown in Figure 1.4 below.

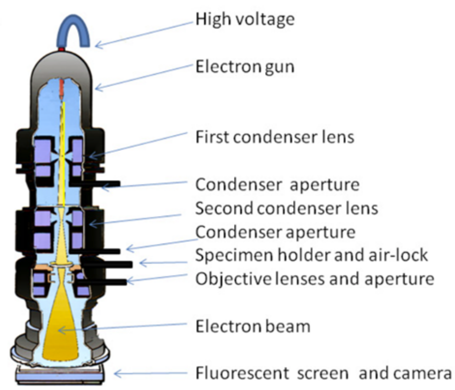

Figure 1.4: Schematic of transmission electron microscope [6]

In contrast to a light microscope, TEM produces higher resolution images because electrons have a smaller wavelength than light. While a light microscope is capable of magnification up to two thousand times, TEM is capable of magnification up to two million times [6].

1.4.2 Hydride Formation

The zirconium alloy Zr-2.5% Nb is used in the pressure tubes of CANDU reactors, which are the pressure boundary and physical barrier of the reactor core [1]. Corrosion of the pressure tubes over time produces deuterium, which accumulates within the alloy. Once the maximum solubility of deuterium in the Zr-2.5% Nb alloy is exceeded, zirconium hydride precipitates can form.

The zirconium hydride is a brittle material which is prone to cracking under stress; this was the cause of pressure tube replacements at the Pickering A station between 1974 and 1976 and has been closely studied since the incident [1]. If a crack forms and penetrates the wall of the pressure tube, it is crucial the reactor can be shut down before full rupture [7]. Monitoring properties such as hydride size and orientation provides insight into the life of the pressure tubes and the velocity of DHC.

1.4.3 Mask R-CNN Model Architecture

Mask R-CNN is an instance segmentation model that extends the “faster region-based convolutional neural network” (Faster R-CNN) framework [3]. Faster R-CNN is a state-of-the-art object detection framework that extracts regions of interest (RoI) from an input image and evaluates convolutional networks on each region to determine class labels [8]. Mask R-CNN adds a branch that predicts masks for each RoI in parallel with the class identification. The Mask R-CNN model architecture is shown in Figure 1.5 below.

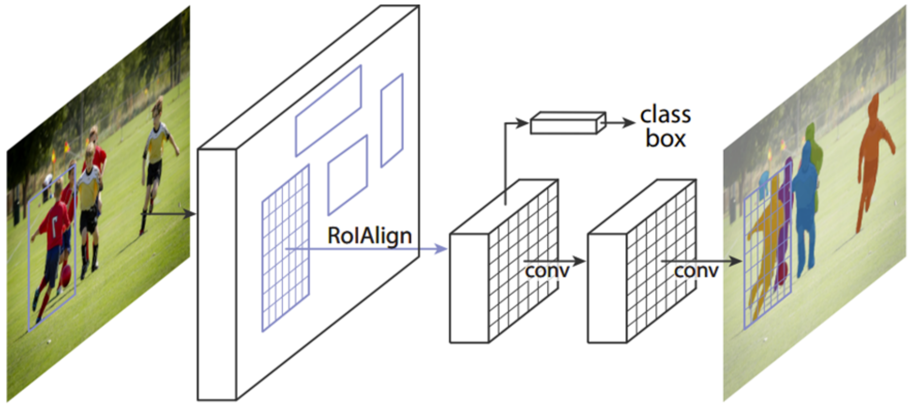

Figure 1.5: Mask R-CNN architecture diagram [3]. Regions of interest are extracted from the original image, then classes are identified in parallel with a convolutional mask generation layer.

The component of the neural network that extracts features is known as the backbone [9]. Mask R-CNN commononly uses a ResNet ("Residual Neural Network") backbone, which passes the image through various convolutional and pooling layers to produce a feature map. The feature map contains abstract information about the image; it maps features of the input image such as lines and edges.

The feature map is then passed into the region proposal network (RPN), which scans the feature map to propose RoI. Finally, the RoIAlign component aligns the RoI with their location in the feature map for class identification and mask prediction.

1.4.4 Software

An understanding of the work previously done by Queen's Computational Materials Science Research Group to implement Mask R-CNN for hydride annotation was important to inform design decisions. This required learning to use several software tools, listed in Table 1.1.

Table 1.1: Overview of the tools and software required to build and deploy the project.

Product Name	Description
Digital Research Alliance of Canada Supercomputers	Supercomputer for training the Mask R-CNN model
MobaXterm	SSH client for connecting to supercomputer remotely
Globus Personal Connect	Tool to transfer files to and from Digital Research Alliance of Canada clusters
VGG Image Annotator	Tool originally used for manual TEM image annotation
VirtualBox	Virtual Machine software
Anaconda Navigator	Allows for set up of virtual environment to run model and create GUI

The VGG Image Annotation Tool is currently used for manual hydride annotation. Practicing image annotation with this tool provided insight into the drawing capabilities the custom annotation tool must provide in addition to the time required to manually annotate images.

The Digitial Research Alliance of Canada Supercomputers provide the necessary compute power to train the Mask R-CNN model. An SSH client, namely MobaXterm, is required to connect to the supercomputer remotely, and Globus Personal Connect is used to transfer files to and from the clusters. The Digital Research Alliance of Canada wiki pages and conversations with Queen's Computational Materials Science Research Group were key resources in learning how to use these [10]. Test jobs using Mask R-CNN were also submitted to the supercomputer to solidify this knowledge.

VirtualBox is a virtual machine (VM) software that would be useful for deploying the final product, discussed in further detail in Chapter 2. Tutorials on using this software, the Ubuntu 22.04 operating system, and the use of Anaconda Navigator for creating virtual environments were followed to prepare for this.

Chapter 2

Preliminary Design

The following contains discussion of the methods used to build the product. This includes developing the annotation tool, deploying the final product to the client, and using revised annotations to continuously improve the Mask R-CNN model.

2.1 Annotation Tool Design

The first step in designing the annotation tool was selecting a Python library for GUI development. The options considered were TKinter, PySimpleGUI, or PyQT [11]. These were compared based on storage requirements, ease of use, and appearance. Storage requirements and ease of use were determined to be the most important criteria as they have the most significant impact on user experience and product success. Appearance was determined to be the least important based on the client's priorities. A score out of two was assigned for each criteria based on a qualitative pass/fail system. The scores achieved by each library are seen in Table 2.1.

Table 2.1: Weighted evaluation matrix created to select a Python library for GUI development.

		TKinter		**PySimpleGUI**		**PyQT**	
Criteria	Weight	Score	Weighted	Score	Weighted	Score	Weighted
Storage Required	50	2	50	1	25	1	25
Ease of Use	30	2	30	1	15	1	15
Appearance	20	1	10	1	10	2	20
Total	**100**		**90**		**50**		**60**

The criteria for assigning scores is outlined in Table 2.2 below.

Table 2.2: Criteria for assigning scores in evaluation matrix.

Criteria	Score 2	Score 1
Storage Required	No additional installation	Must be installed separately
Ease of Use	Numerous relevant tutorials available online	Difficult to find relevant tutorials
Appearance	GUIs built with the library are visually pleasing	GUIs built with the library have a standard appearance

TKinter was selected as the library to be used for the annotation tool because it achieved the highest weighted score. It is the default GUI library in Python and considered to be the recommended library for tool GUIs, and therefore is the most used in relevant tutorials [11].

Another option considered was using the VGG Image Annotation tool for re-annotation rather than creating a new tool, however this would not provide a seamless user experience and therefore was not pursued.

2.2 Product Deployment

For ease of use, a VM was chosen as the method of deployment. This allowed the product to be designed and tested, then saved and securely shared in its fully functional form with the CNL team. Once screened by their IT team, the tool can be imported directly into a virtual machine software and used immediately.

The most widespread VM technologies are VirtualBox and VMware. VirtualBox was chosen due to its simple installation and setup, which will provide a strong user experience [12].

2.3 Mask R-CNN Model Iteration

Once a collection of re-annotated images is collected, these can be used to re-train the Mask R-CNN model to continuously improve its performance. The revised annotations must be saved by the annotation correction tool and shared from the CNL team to the AI engineer via secure file transfer protocol (SFTP). The model can then be re-trained on the Digital Research Alliance of Canada supercomputer.

2.4 User Experience

The user flow diagram in Figure 2.1 below models the process for the end user to use the tool.

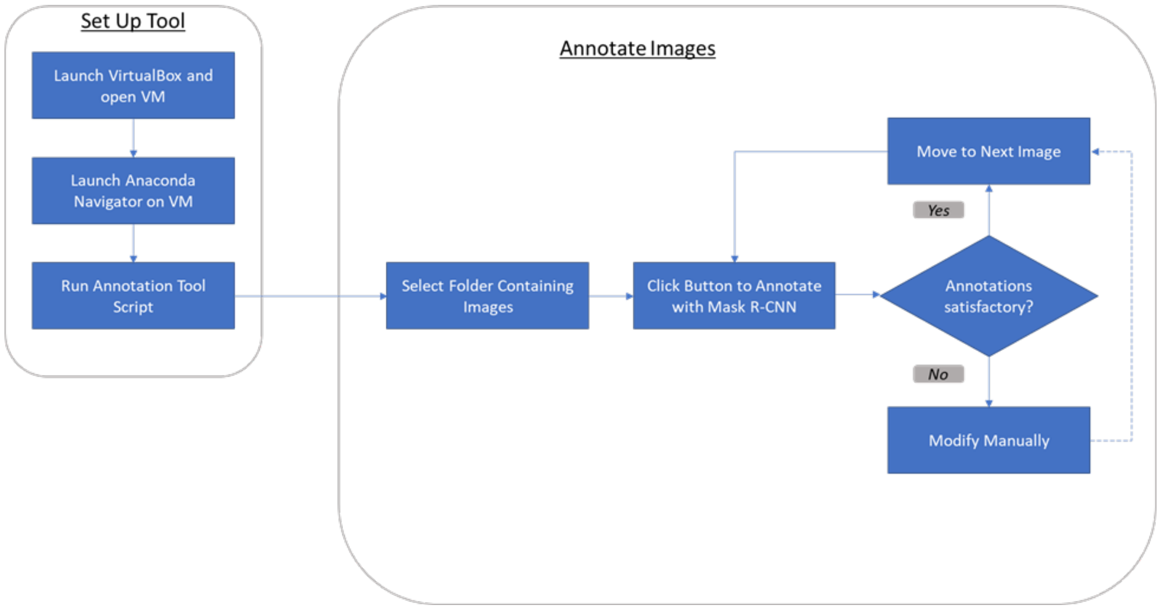

Figure 2.1: User flow diagram for the annotation tool.

To set up the tool on a day-to-day basis, the user would open the VM on VirtualBox, launch Anaconda Navigator, and run the annotation tool script to open the tool's GUI. From there, they select the folder containing images and click a button to automatically annotate the current image with Mask R-CNN. If the annotation is satisfactory, they may navigate to the next image, otherwise they can manually modify the annotations before proceeding.

Chapter 3

Product Development and Iteration

The following contains details surrounding product development, testing, and iteration to ensure the tool met the client's needs. An annotation tool was first developed in Python using Tkinter, which was then integrated with the Mask R-CNN model and deployed to CNL using a VM for testing and iteration.

3.1 Annotation Tool Development

An iterative approach was taken to achieve the desired functionality for the annotation tool. Several tutorials on effective UI development were followed to build a foundational understanding of the capabilities of Tkinter, then extended to create the tool [13]. Three design iterations are seen in Figure 3.1 below.

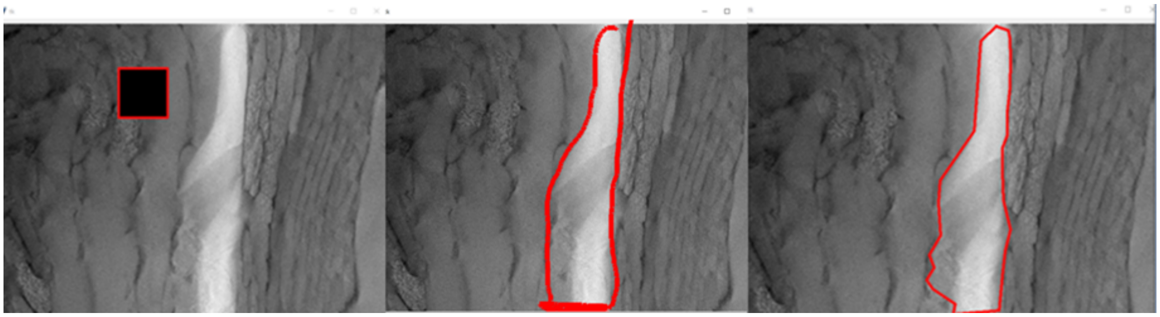

Figure 3.1: Three design iterations of the annotation tool. From left to right, a tool that displays a predefined shape, a tool that allows users to draw freehand by dragging the mouse, and a tool that allows users to draw clear polygons on the canvas.

The simplest design iteration was an image bound to a canvas, with a pre-defined shape as an annotation. The next stage added functionality to draw freehand on the canvas [14]. Finally, this was extended to build a new tool allowing users to draw custom polygonal shapes on a canvas, similar to the VGG Image Annotator currently being used by the team [2]. Each click of the mouse is stored as a vertex, and the shape is displayed in real time as the user draws.

3.2 Integration with Mask R-CNN

The Mask R-CNN model was then integrated with the annotation tool. The user selects the folder containing images and navigates through them one by one. By clicking a button, images are annotated with Mask R-CNN and can be revised by drawing a polygon directly on canvas. This works for images with a single hydride instance in addition to images with multiple instances. Any new annotations are saved to a CSV file. A screenshot of the interface is seen in Figure 3.2.

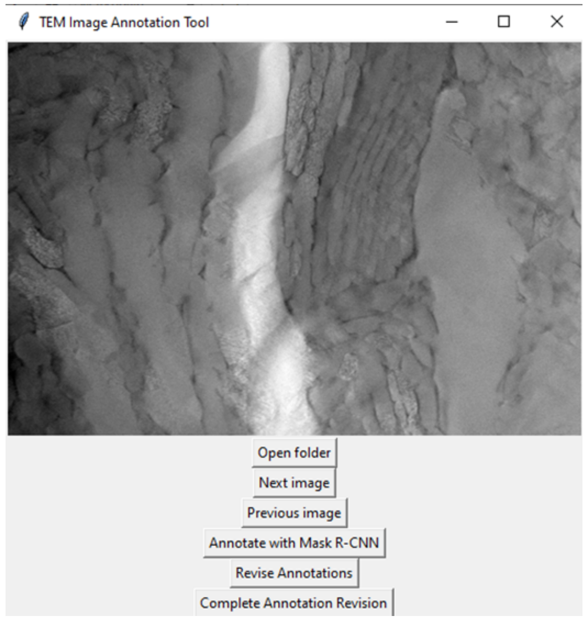

Figure 3.2: Screenshot of UI for first iteration of annotation tool.

3.3 Product Deployment

A VM was configured with Ubuntu 22.04. A virtual environment with the required libraries was then set up using Anaconda Navigator to allow for the use of Mask R-CNN. Guest additions were also installed to improve VM performance [15].

The annotation tool, which was developed locally for convenience, was transferred from local storage to the VM using Globus. The VM was then exported in OVA format and uploaded to CNL server via SFTP using FileZilla. Finally, a CNL stakeholder downloaded the OVA file and imported it into VirtualBox to start using the annotation tool.

The VM itself occupies 25 GB of storage and the compressed OVA file used to share the VM with the client occupies 15 GB. This takes approximately 50 minutes to upload to the CNL server at an upload speed of 5 Mbps and approximately 5 minutes to download a download speed of 50 Mbps , though these values will vary depending on the user's internet.

3.4 Validation, Testing, and Iteration

Progress was shared on an ongoing basis through a biweekly meeting with CNL and Queen's University stakeholders. These frequent touchpoints ensured the product stayed on track to meet the client's needs. Feedback received during these meetings throughout the year was generally positive, and no major pivots were required.

Once the first iteration of the tool was complete, it was tested with a variety of images both on the host and the VM to ensure it was performing as expected. Images can be annotated with Mask R-CNN in approximately 10 seconds on the host and 40 seconds on the VM. This difference is significant, but still within the original 1 minute per image time constraint. It is likely due to inadequate host hardware [15].

I then worked with one of the CNL stakeholders to test the product on a CNL host and improve the design. The client was pleased with the design and ease of use overall. They suggested implementing an "annotate all" feature that would allow the user to annotate all images in a folder at once rather than wait for each individual annotation. They also pointed out that although modified annotations were saved by the tool, automatic annotations were not, and should be to provide full functionality.

The recommended "annotate all" feature was implemented to reduce the inconvenience of the Mask R-CNN annotation lag. Future testing includes adjusting the CPU and RAM allotment to the VM to improve the speed of annotating images on the VM [15].

Chapter 4

Final Design

The suggestions made by the client to improve the usability of the tool were applied to create the final product.

4.1 Annotation Correction Tool

The UI of the final iteration of the annotation tool is shown in Figure 4.1 below.

Figure 4.1: Screenshot of UI for final iteration of annotation tool.

A breakdown of the attributes and methods of the AnnotationTool class is modelled in Figure 4.2 below.

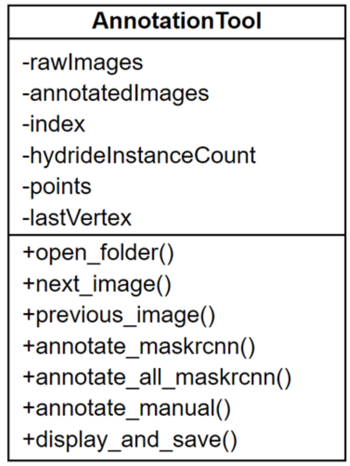

Figure 4.2: UML class diagram for the AnnotationTool class. The methods are listed at the bottom and the attributes are above.

The annotation tool class contains attributes for the raw images in addition to any annotated images, the index of the current image being viewed, and the number of hydride instances on manually annotated images. It also has an attribute containing revised annotations in format [imageName, instanceNum, x1, y1, x2, y2, x3, y3 ...] and the last x-y coordinate of the mouse for drawing annotations. It contains methods to open the folder, navigate through images, annotate one or multiple images with Mask R-CNN, manually annotate images, and save completed annotations. Each method is bound to a button on the UI.

4.2 Economic and Environmental Analysis

In order to train the Mask R-CNN model, compute power from the Digital Research Alliance of Canada supercomputers is required. Membership fees to use these resources cost between \$5000 and \$15,000 annually depending on research intensity [16]. The annotation correction tool itself uses entirely free and open-source software. Therefore, once the image segmentation model is trained, the cost to use the final product is negligible. There is also no cost to scale the tool for use in other applications, other than the time for development.

The more this tool is used, the more revised annotations will accumulate which will be useful for re-training the model and improving accuracy. However, it is important to consider the environmental impact of training the Mask R-CNN model due to energy consumption. To reduce the amount of compute power required, the frequency of re-training should be reduced where possible.

Chapter 5

Conclusion

A tool that allows CNL microscopists to easily interact with the Mask R-CNN model and revise annotations independently was successfully created and implemented. Images can be annotated automatically using Mask R-CNN with the click of a button, and annotations can be modified by drawing on the image.

The completed VM can be uploaded to the CNL server within 50 minutes and downloaded by the microscopist within 5 minutes. An acceptable VM size was not quantified at the start of the project, however the client feels the times achieved are satisfactory. All files were shared using SFTP and screened by CNL's IT team for approval.

The time to annotate a single image on the VM is 40 seconds, which meets the 1 minute per image processing time requirement. The inconvenience of this delay was mitigated by creating an "annotate all" option that allows this task to run in the background. Annotation turnaround time is estimated to be reduced to one day.

The product requires one additional feature that integrates automated and modified annotations into a single CSV output to provide full functionality. This was not identified in the original scope and will be completed prior to the end of the semester. The tool can be used both for annotation of hydride datasets and live during TEM image collection; this will allow the microscopist to see how the Mask R-CNN model is performing in real time and adjust contrast as needed to reduce manual analysis later.

5.1 Next Steps

The tool can be improved by optimizing the performance of the VM itself. VM performance is directly related to CPU and RAM allotments, which were minimized to reduce the size of the VM. A investigation of the trade-off between VM size and product performance would be beneficial.

A plan should also be developed for increasing the storage of the VM if required. Currently, the VM has 1 GB of storage remaining, which may need to be increased for large collections of images. Potential plans include simply increasing VM size to eliminate this issue, or writing a guide for CNL stakeholders to increase VM size independently as required.

Future work includes extending this tool to the annotation of grain boundaries. This new application in addition to the other proposed next steps will be worked on in the remaining weeks of the semester. New iterations of the tool can be easily shared by uploading the OVA file to the CNL server.

Bibliography

[1] W. J. Garland, "The essential candu: a textbook on the candu nuclear power plant technology: University network of excellence in nuclear engineering," 2016.

[2] A. Dutta, A. Gupta, and A. Zissermann, "Vgg image annotator (via)," *URL: http://www. robots. ox. ac. uk/˜ vgg/software/via*, 2016. Accessed 2023-02-08.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[5] "Transmission electron microscopy (tem)." `https://www.nottingham.ac.uk/nmrc-commercial/nmcs-facilities/tem.aspx`. Accessed 2023-03-30.

[6] "The transmission electron microscope." `https://www.ccber.ucsb.edu/ucsb-natural-history-collections-botanical-plant-anatomy/transmission-electron-microscope`. Accessed 2023-03-30.

[7] I. A. E. Agency, *Delayed Hydride Cracking in Zirconium Alloys in Pressure Tube Nuclear Reactors. IAEA TECDOC Series No. 1410.* International Atomic Energy Agency, 2004.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[9] "Understanding mask r-cnn basic architecture." `https://www.shuffleai.blog/blog/Understanding_Mask_R-CNN_Basic_Architecture.html`. Accessed 2023-04-01.

[10] "Technical documentation." `https://docs.alliancecan.ca/wiki/Technical_documentation`. Accessed 2023-03-31.

[11] M. Fitzpatrick, "Which python gui library should you use in 2023?." `https://www.pythonguis.com/faq/which-python-gui-library/`, Mar 2023. Accessed 2023-03-30.

[12] "Vmware vs virtualbox: What's the difference?." `https://www.interviewbit.com/blog/vmware-vs-virtualbox/`, Oct 2021. Accessed 2023-02-08.

[13] "Tkinter - python interface to tcl/tk." `https://docs.python.org/3/library/tkinter.html`. Accessed 2023-02-08.

[14] "Tkinter python gui tutorial for beginners - create simple paint application using tkinter." `https://www.youtube.com/watch?v=kzp7-0EFrIg`, Jun 2019. Accessed 2023-02-13.

[15] NishantGola@TWC, "How to speed up a virtualbox virtual machine and make it run faster." `https://www.thewindowsclub.com/speed-up-a-virtualbox-vm`, Nov 2021. Accessed 2023-04-01.

[16] "Become a member." `https://alliancecan.ca/en/membership/become-member`. Accessed 2023-03-31.

Appendix A

Statement of Work and Contributions

The Computational Materials Science Research Group at Queen's University implemented and trained the Mask R-CNN model prior to my commencement of the project. They helped me get up to speed on how this was done and how the model can be re-trained in the future if required.

I divided my work into background learning and research in the fall term and implementation in the winter term. I spent Monday mornings from 9 am until 3 pm in both terms working on this project in Nicol Hall and alternating Tuesdays from 10am until 11 am meeting with Queen's and CNL stakeholders. Table A.1 shows the timeline followed, with tasks from original plan that were determined unnecessary marked N/A.

Table A.1: Summary of work completed in the fall and winter terms.

Task	**Target Deadline**	**Date of Completion**
Attend biweekly meetings with CNL team; communicate findings and seek feedback	Ongoing	Ongoing
Background research on U-Net and Mask R-CNN architecture, create Mask R-CNN environment in anaconda and use trained model to produce annotations	Oct 17	Oct 17
Background research on supercomputers and submitting jobs in Linux	Oct 24	Oct 24
Create Mask R-CNN and U-Net environments on Compute Canada cluster and submit an image annotation job successfully	Nov 7	Nov 7
Set up virtual machine using VirtualBox with Ubuntu 22.04 operating system, create Mask R-CNN environment in anaconda on VM and use trained model to produce annotations	Nov 14	Nov 21
Create annotation tool in Python	Jan 16	Jan 20
Integrate annotation tool with Mask R-CNN network and annotations	Jan 23	Feb 10
Fine tune user interface	Feb 6	Feb 13
Test product by transferring virtual machine to new host in Nicol Hall	Feb 13	N/A
Design iteration	Feb 27	N/A
Deploy product to CNL team and create guide	Mar 6	Feb 17
Design iteration	Mar 20	Ongoing
Implement new tool features with remaining time	April 10	April 10

Appendix B

Acronyms

Table B.1 contains information on commonly used acronyms in this report.

Table B.1: Description of commonly used acronyms.

Acronym	**Description**
TEM	A Transmission Electron Micrograph (TEM) is the type of image being annotated
CNL	Canadian Nuclear Labs (CNL) is the client
DHC	Delayed Hydride Cracking (DHC) is a corrosion issue in nuclear reactors that we aim to study
Mask R-CNN	Mask Region-Based Convolutional Neural Network (Mask R-CNN) is the neural network used for hydride annotation
VM	A Virtual Machine (VM) was used to deploy the product to the client
GUI	The product has a Graphical User Interface (GUI), providing functionality through buttons
SFTP	Secure File Transfer Protocol (SFTP) is used for securely share files between Queen's University and CNL