Final Documentation

UTD CISI: Deep Learning AI Tools for Medical Imaging

Team Members: Subhayan Basu, Sujay Karanam, Justin Jung, Paul Ko, Aryan Patel, and Leo Kuo

ABSTRACT : Our project deals with implementing Deep Learning concepts, such as U-net models, and different programming languages to create a product that would help Medical Imaging, specifically human placental imaging, and ultimately help medical professionals as much as possible. This project aims at segmenting the uterine cavity and placenta in fetal MR images to detect abnormalities affecting maternal and fetal health. The project entails developing a web application, a flywheel plug-in, and a desktop application based on the research of uterine and placental segmentation in MR images for the early detection of health issues. Incorporating the neural network algorithm from the study into applications with user-friendly interfaces allows medical professionals to access this technology more easily.

TABLE OF CONTENTS

I) **INTRODUCTION**: The most obvious factor behind our project being unique is that it uses Artificial Intelligence to make an impact in medical imaging. For many years, the thought of AI impacting the medical world has been in the minds of millions of people, and it has made a good impact so far. It's only the beginning though, because it is predicted that AI's influence in medicine will keep increasing. Artificial Intelligence itself has many different fields, such as Machine Learning, Deep Learning, Computer Vision, and more. Every single aspect of AI, whether very small or large, helps push AI to reach greater heights. Our project specifically focuses on Segmentation of the uterus and placenta in MRI. Another unique thing about the project is that we make use of the U-net Convolutional Neural Network (CNN). U-net is a type of CNN capable of making a greater impact than some other types of Neural Networks, especially those that aren't used much nowadays and ones that have more limitations than U-net.

While there are many ways to create deep learning neural networks to solve the issue of segmentation of the uterus and the placenta in MRIs, U-NETS have been recognized to work very well. The problem with medical imaging concerning the deep learning algorithm that uses the U-Nets to segment the placenta from the uterus is that while it can prove to be extremely useful for health specialists, there aren't many ways to use the algorithm. Our team's primary objective is to make this helpful algorithm easier to access and use this algorithm to help better diagnose and identify any problems that may arise in a pregnant woman. There are multiple solutions to effectively make this algorithm easier to access; for example, the algorithm could be published on a popular research website to help garner attention, but this doesn't stop the fact that the algorithm itself is still not easily applicable. There are other ways to effectively make the algorithm more accessible such as building a GUI or web application. The problem that arises from building just one of these things is that by just building one of these tools, we severely limit the potential for accessibility by the research and medical experts. Instead, our team will build a web application, a GUI, and a plugin utilizing flywheel. By building these tools, we can then cater to the accessibility of the algorithm to many people since these tools have a wide range of platforms.

TIMETABLE

Phase 1: Wed 10/25/23

Web application

- (Due 10/18) Have the guidelines and details for all features on the Web system
- (Due 10/25) Have the wireframes completed, with a working UI

GUI

- 10/18
    - We figured out the basics of the GUI and how exactly it will fit in and impact the project
    - We successfully had our IDE's working on our computers so that it would enable us to easily work on Python
- 10/25
    - We read the research paper Dr. Fei gave us to understand the model better

FLYWHEEL UI

- (Due 10/18)

    - Researched and understood the required documents and features to build the FLYWHEEL UI such as the run file, docker, and manifest file.
    - Understood the UNET model on segmentation of the MRI on the Placenta and Uterus.
    - Read the research paper associated with the segmentation model to better understand it and learn how to run the model correctly.
    - Learned how to open and integrate dicom files into Python code

- (Due 10/25)

    - Figured out the model input shapes and output shapes along with other details about the model and the research that was done by it.

- Ran a sample code by stacking the dicom images into the input shape to get a sample output from the model to better understand the output format.

Phase 2: Mon 11/20/23

Web application

- (Due 11/13) Have a working prototype of the web system which will need further changes proposed from sponsor feedback

GUI Application

11/15:

- We made good progress on creating the GUI and finished almost all the code for it

11/20

  - Got the GUI to display images, however, they were not all correct
  - Got a lot of the code well-written

FLYWHEEL UI

- (Due 11/13)

  - Researched more on the FLYWHEEL UI and understand better the creation of the flywheel gears
  - Started and finished creating Docker file to the best of our ability
  - Studied on how to create JSON files and built a Manifest file to the best of our ability
  - Started the pre-processing of data by first getting the dicom images to the right size
  - Normalized and truncated the voxel distribution of the 2d slices of the DICOM IMAGES
  - Post processed the output into shapes of 255,255 to be able to output to screen.

- (Due 11/20)

  - Got feedback from the model's output and went back to add changes with the pre and post-processing
  - Applied a median filter on the 2D slices in the pre-processing

- Corrected the way the code was blocking the dicom slices for the input and reworked the code to correctly block the slices in an overlapping fashion
- Corrected the steps that were taken to get the blocks into the shape that was necessary for the model.
- Applied preprocessing by getting the probabilities into a shape that would be easier to manipulate and use
- Applied averaging of the overlapped slices to get the original number of slices that was initially inputted.
- Applied a threshold of 0.5 to get the binary map for each channel and created a mask which was applied to the original slice

Phase 3: Fri 12/8/23

Web Application

- Created API for the backend
- Able to upload and delete selected DICOM files
- Working home page

GUI Application

- Corrected the issues we were having with the GUI and got it to work completely well
- Outputted the original image, and the 3 masked images that relate to the original image
- Put in some more comments to the code

FLYWHEEL UI

- (Due 12/8)

- Corrected the postprocessing by removing the averaging of the slices in the postprocessing section and instead treated each block of slices separately and outputted them seperately.
- Outputted the overlapped slices separately with the corresponding mask applied to it
- Generalized the code, added comments, and formatted it neatly to be better understood by a reader

-

PROJECT METRICS

In order to have the success of the project, our team decided to use user surveys and feedback from customer support to collect feedback and identify any issues and shortcomings. Specifically, the users would be the doctors and surgeons as they would have to use it on their patients. To make the most accurate conclusion of the application's integrity, we would like to see how satisfied the doctor was with our project, how beneficial it was, and what possible drawbacks and errors there were. We plan on doing a peer review across all members of the team by asking each member what their opinions are on the project. Specifically, we'll get their thoughts on potential improvement ideas and potential drawbacks that may occur. Since our project deploys an algorithm, we will write code that tests the efficiency of the algorithm. So analyzing it will help us understand how we can improve it by making it faster. Additionally, analyzing the algorithm may lead us to fix potential errors.

KEY ROLES

- Justin Jung is the team leader. He will be coordinating meetings, and time management with the group, and will be submitting documents for the team as a team leader.
- Aryan Patel, as an assistant team leader, will be counting the minutes times with the team and will write down what was said in the meeting, which will help the team understand what is done and what needs to be done.
- Leo Kuo and Justin Jung will be creating the web application that consists of the landing page where images/files are uploaded and the confirmation page were the uploaded files will be viewed.
- Subhayan Basu and Aryan Patel will be creating and optimizing the GUI. They will be displaying important images of the model in the GUI.
- Sujay Karanam and Paul Ko will be implementing and optimizing the deep learning algorithm to be used in a software plugin for Flywheel, a medical imaging AI development platform.
- Dr. Baowei Fei will be the one to mentor us through this process and be our point of contact. As a highly respected and skilled individual, the team will be asking questions and will try to understand all that our sponsor tasks us to fulfill the needs of Dr. Baowei Fei.

IMPLEMENTATION DETAILS

Web Application:

For the Web application part, we used modern frameworks like Nextj.js for server-side rendering and SEO benefits. For the landing page of the website, there is an area for users to drop or select files to upload and a submit button to send the files to the backend. After users select the files to upload, they can edit the files. Then the website will link to the confirmation page. In the confirmation page, users can view the result of the DICOm images. To integrate the frontend and backend, we used Flask to create the API. The frontend will send andDICOM images and receive output through the API.

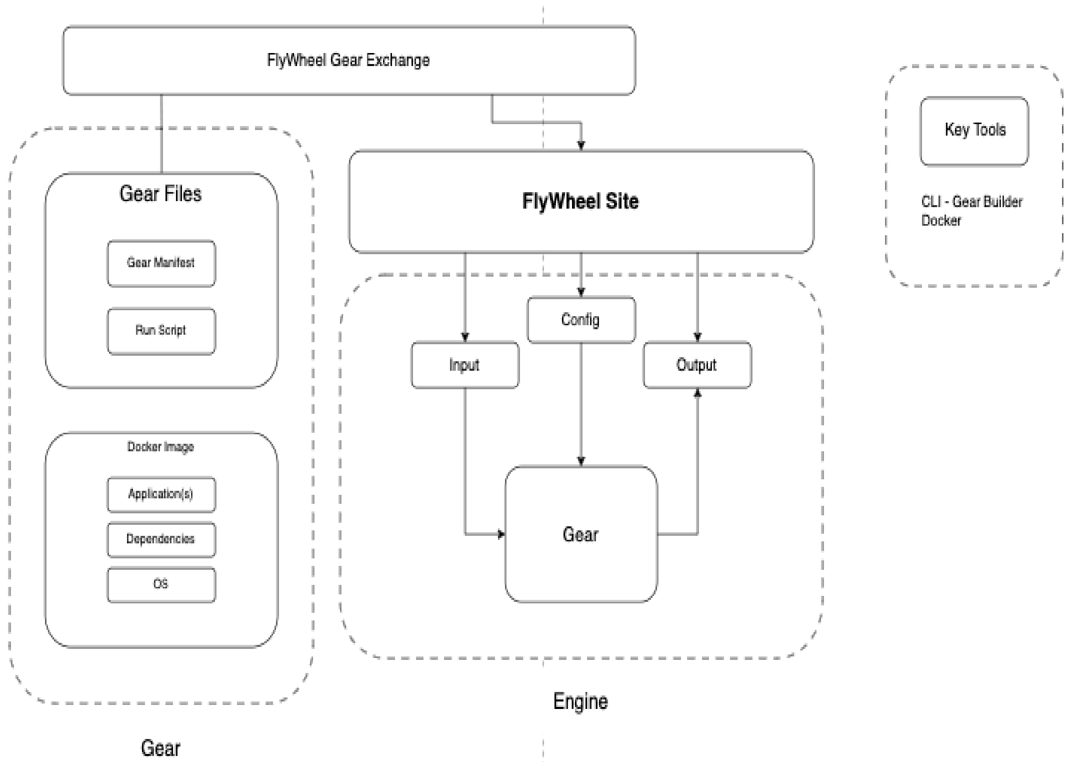
Gui:

For the GUI, we used a variety of resources that python provides. We used tinker to set up the window, with all the different widgets. Some of the widgets are the next, previous and slider bar. These widgets then need to be programmed to traverse through the dicom images so that the user is able to view correctly. We then needed to implement the model code with the GUI itself. For this, we used a plt save function to save all the images in the working directory and then uploading them back into the GUI where we could actually view them.

FLYWHEEL UI:

For the model, we had a lot of use cases. The first use case we used was that of a single block of 5 slices to see if the model itself was working. Then the second use case was that of colored slices to check if the grayscale and median filter were working. For the third case, we then expanded the input size to a whole 3d DICOM image to make sure the model can run on a whole DICOM image and output in the correct format. To then test the post-processing steps and functions that rewrite the arrays into data structures that can be outputted onto the screen, we went into the original DICOM image and spliced it up creating sub-images of different sizes and inputted them into the code to test that the code is generalized and works for all input and input sizes.

Architecture Design:



COMMUNICATION PLAN

The way we plan to communicate with our teammates is with the use of Discord. Within Discord we have set certain channels such as weekly guidelines and voice channels in order to communicate with each other efficiently. We have created a GitHub project in order to better organize and help our team stay synced with one another. In addition to Discord, once we start working on the project we plan to meet in person during our class times, so we can work on the project together and receive input from one another directly. The way we interact with our sponsor is by the use of Microsoft Teams, where we have created a group chat to help set up meetings with our sponsor and to ask our questions in that room. We meet with our sponsor at the Bioengineering and Sciences building on campus weekly, where we talk about milestones and the next steps for the projects. Outside our weekly meetings with our sponsor, our team also meets on Discord or in person via class time and also 3 other times within the week. We set up when2meet links in order to find a time that works for most people. The way we keep track of attendees is by creating a document called meeting minutes that is sent to our sponsor after every meeting. The document will mark who was attending the meeting as well as the action items, stating what we discussed and also what needs to be done before the next meeting. This helps our team to stay on track and communicate efficiently.

RISK ANALYSIS/CONTINGENCY PLAN:

Formulating a risk analysis and contingency plan is critical for project success. Team redundancy can be achieved by having diversified skill sets in case anyone on the team is absent unexpectedly. Everyone on the team has been learning python and tensorflow since the beginning and knows basic knowledge of front-end development. Identifying alternative software and equipment solutions are achieved by considering using the following software such as using Discord or Microsoft Team. Our team also decided to implement regular team check-ins for early issue detection and resolution. Cross-training team members for Python and tensorflow enhances versatility. Additionally, our team has outlined a clear escalation procedure, encompassing incident identification, initial resolution within the team, secondary to mentors, documentation of the incidents, and keeping everyone on the team informed through a communication plan. If a team member is sick or if we believe they have gone missing, then we would immediately report our project mentor and faculty advisor regarding it.

EVALUATION/TRACKING PLAN:

The way we would know by the end of the semester that the project was successful is by having a clear-cut plan of our project and all the objectives that would be needed to finish the project. By having weekly meetings with our sponsor, it allows us to have our current progress and work assessed in a weekly manner. This allows us to have a valid evaluation plan by checking in with our sponsor weekly to analyze our work and progress. The sponsor will look at our work and progress and give us feedback and the upcoming tasks which will allow us to stay on track on the project and present a project that will satisfy our sponsor. On the due date of every phase, we as a whole team will evaluate our progress by checking the work the 3 subgroups have done. Our planned due dates are October 25th for Phase 1, November 13th for Phase 2, and December 5th for Phase 3.

ETHICS DISCUSSION:

Splitting the work in the group has been done pretty well as there are 3 subgroups with 2 people each, and every person is contributing well to their subgroup. Both our team leader and assistant team leaders have been doing their roles well and have been giving us a lot of respect. Ensuring fair work distribution among the team fosters a collaborative environment. Additionally, the code's origin must be considered. Ethical sourcing of code, especially from non-open source platforms, demands adherence to agreements to avoid legal and ethical violations. We will not be using code from other people or any other sources, unless it is code that is commonly used by others that can be beneficial to our project, without having academic honesty. Lastly, this project emphasizes utilizing and leveraging technological advancements in medical imaging. The handling of sensitive medical images requires strict confidentiality protocols, which will be established by signing Non-Disclosure Agreements. This legal framework enhances safeguarding privacy and aligns with ethical principles.

IMPACT, SECURITY, AND PRIVACY:

Our project can advance the medical imaging sector of UTD CISI and eventually make a mark on the medical field. More importantly, the project would hopefully make it easier for all doctors to make clinical applications of human placental imaging. Specifically, the project can enhance the detection of abnormalities that affect maternal and fetal health. We will try to keep our application as secure as possible by giving only some special doctors access to it. To maintain confidentiality, the application will never have the names of any patients who were treated using the app. Anyone who uses our application will have to create an account and enter their password to use it.

INDIVIDUAL ASSESSMENT:

Subhayan Basu:

As a member of the GUI subteam, I contributed a pretty good amount and had good collaborations with my teammate Aryan. When I first started implementing the GUI, I practiced writing Python code on how to crop an image. I also practiced Python coding of working with Tensorflow. Soon enough, I wrote code for a user login page, and a “Create your account” page. Those things were not used in the GUI though. During the time between the end of October to the week before Thanksgiving break, I made some contributions to the model’s code (the one that runs the U-net model and displays tons of images). What I mostly did on that part was test the code and make a few changes to make the output more accurate and reduce the chances of errors. For the GUI, I spent a lot of time trying to make the code output the correct images as it was not easy at all. My partner and I encountered a roadblock on the GUI during Thanksgiving break where the GUI was not displaying any images. I changed the code and added a bit more to make it display images, however, the images getting displayed were not correct. To fix that issue, I planned on incorporating the codes of the model and the GUI together so that all the correct images on the GUI could show. Finally, in the end, my partner and I got the GUI running well. Working on the GUI was a great experience for me and I learned alot from there.

Justin Jung: In my role within the web application project, I was tasked with implementing crucial functionalities that significantly enriched the user experience. One pivotal aspect of my contribution involved coding the image preview feature, allowing users to preview their uploaded images before confirming their actions. This required in-depth knowledge of front-end technologies, particularly JavaScript and relevant libraries, to enable dynamic and real-time image rendering on the user interface. Additionally, I undertook the challenge of converting DICOM images into JPEG files to ensure seamless viewing on the confirmation page. This process involved coding a robust conversion mechanism, navigating the complexities of DICOM file structures, and employing image processing techniques. Furthermore, by researching how to connect the U-net model to the application required skills like flash and fastapi while also learning about neural network models. By successfully researching and learning these features, I not only enhanced the overall functionality of the web application but also deepened my expertise in image manipulation and file format conversion within the context of web development.

Paul Ko: As a member of the flywheel team, I played a significant role in tackling the challenges associated with the Flywheel plugin project and the U-Net model for image segmentation while having the pleasure to work with my subteam partner, Sujay. For the U-Net model segment, I demonstrated a deep understanding of the research paper and successfully grasped the intricacies of the model input and output shapes, along with other critical details. By running a sample code to process DICOM images, I gained valuable insights into the model's output format. I also contributed significantly to the research phase, delving into the FLYWHEEL UI and making strides in comprehending the creation of Flywheel gears. In the first phase of the Flywheel project, my partner and I took on the responsibility of enhancing the U-Net model's capabilities by integrating it with the run file for the Flywheel plugin software. This involved a meticulous understanding of DICOM image analysis and leveraging Python for the integration. For the second phase, focusing on the Flywheel UI, I worked collaboratively to create essential files like the dockerfile, run file, and manifest.JSON file, despite being relatively new to JSON format. Overall, I found this to be quite a challenging experience. As of the current status, I have successfully researched and understood the required elements for the Flywheel UI, including the creation of Flywheel gears. The creation of the Manifest file and Docker file were completed in a timely manner by the deadline of November 13th, and I am looking forward to adding additional features to the gear for the commercialization. In conclusion, my proactive approach, solid technical skills, and ability to adapt to new technologies have contributed to the success of the project. While some tasks were completed, I was able to help the subteam and ensure the successful implementation of the model into the flywheel UI, further enhancing my skills in Python, deep learning models, and software development.

Aryan Patel:

I was tasked with the GUI part of the project. With my partner Subhayan, we were able to get a working finalized product where we were able to view the model output of the DICOM images in a nice and clean GUI system. The system was built in python and we had many issues going through this project. My team and I had done a lot of research and figured these out together. Contributing to the development of a GUI system using Python and Tkinter was a challenging yet rewarding experience. As a team member, I encountered various obstacles, particularly compatibility issues that arose during the implementation phase. Working diligently, we addressed these challenges head-on, leveraging our problem-solving skills and collaborating seamlessly to find effective solutions. The Tkinter framework, known for its simplicity and versatility, served as a solid foundation for creating a user-friendly interface. Despite initial hurdles, the team's commitment and expertise prevailed, leading to the successful resolution of compatibility issues and the eventual completion of the project. This experience not only honed our technical skills but also highlighted the importance of perseverance and teamwork in overcoming software development challenges. The final GUI system works well and shows the correct output.

Leo Kuo:

I was assigned to the Web application part of the project. Working on the project alongside Justin was an enriching experience. As a member of the team, my primary responsibility was to build an interface that is user-friendly, intuitive, and efficient for medical professionals. I dedicated myself to build a user experience that simplified the complexity of AI tools ensuring that the application was straightforward and easy to use. For the backend part of the application, I was not familiar with python and neural networks. Therefore, I spent a lot of time researching the preprocessing and postprocessing part of the U-net model. My research involved studying research papers given by Dr. Fei. For the implementation of connecting the backend and frontend, I did a lot of studying on how to create api with Flask and Fastapi and input the user file into the backend. I also spent a lot of time researching how to render DICOM files on browsers and unfortunately I failed to do this feature. Through this project, my expertise in Next.js and python was greatly enhanced. I gained a deeper understanding of AI and neural networks. This experience taught me not only about coding and design, but also the practical application of AI in healthcare. The collaborative effort with Justin was important in overcoming challenges and achieving our project goals. This project allowed me to apply my skills in a meaningful way.

Sujay Karanam:

I was assigned to the FLYWHEEL UI team alongside Paul. As a member of the team, my responsibilities were to help create the FLYWHEEL UI and also help with the deep learning UNET model. I was the main frontrunner working on the deep learning model along with some contributions from my teammates. My main role was to figure out how the model worked and how to implement it. I also created the main run file that the mode was implemented in so the inputs and outputs, postprocessing, and preprocessing were formatted into the code. I worked on the preprocessing, the postprocessing, and the outputting of the model's predictions. I also worked on the manifest and the main run file needed for the FLYWHEEL UI. The completed tasks that I have done or contributed towards that were part of phase 1 and due on October 18 were researched on how the FLYWHEEL UI worked and what components were needed to create the UI, research and understand the UNET model, and the details about it, and get familiarized with the structure of the DICOM file format and know how to manipulate them for usage in Python. The completed tasks that I have done or contributed towards that were part of phase 1 and due on October 25 were to better understand the research behind the model and figure out more details, figure out the input shape and output shape, and run a sample code that provides more insight into how the output of the model is formatted. The completed tasks that I have done or contributed towards that were part of phase 2 and due on November 13 were understanding the creation of the flywheel gears better, creating the docker file, processing the input DICOM files, and postprocessing the output from the model. The completed tasks that I have done or contributed towards that were part of phase 2 and due on November 20 were reviewing the previous code and applying more preprocessing and postprocessing techniques, correcting previous code and replacing it with more correct code, adding code that desirably outputs the results. For the last phase of the project, I am currently working on the model to get better results by going back to the post-processing step, reviewing some of the averaging functions I created, and trying to post-process the individual blocks separately rather than treating them as overlapped.

ISSUES AND LESSONS LEARNED:

Subhayan:

I definitely learned more Python through this project. Before doing this project, I had previous knowledge of Python from what I learned in my Machine Learning class last semester, but it was definitely not enough for this project. I encountered several issues while working on this project. The very first issue I encountered was being able to work on the IDE PyCharm. At first, I was lost and panicking that I wouldn't be able to work on that IDE, but I realized that there were different Python interpreters I could use on the IDE. Some of the packages (like tensorflow, cv2, and pydicom) were not getting installed in the IDE, and I spent hours trying to figure it out. But when I realized there was a different Python Interpreter, it made it easy for me to get the python packages running and start working on the GUI. Another issue I encountered was how to code in Python using some of those libraries, like Tensorflow, keras, and pydicom. It took a while to understand how to code using those things in Python before I could begin working on the GUI. The biggest issue I encountered with the GUI was how to write code for reading and uploading images onto the GUI. After that came the issue of the GUI to display the correct images. I was happy it all worked out at the end and this project taught me well.

Justin Jung: Through my involvement in the web application part of the project, I acquired valuable skills in python, javascript, html, and css. One key learning was the implementation of image preview functionality after users uploaded their files. This required a deep understanding of front-end development, including JavaScript and related libraries, to create an interactive and seamless user experience. Additionally, a significant aspect of my learning journey involved the conversion of DICOM images into JPEG files to facilitate their display on the confirmation page. This process not only demanded familiarity with image processing techniques but also required a grasp of the DICOM format and the intricacies of converting it into a widely supported format like JPEG. Overall, this project provided me with a comprehensive understanding of image manipulation in web development, enhancing my technical proficiency and problem-solving skills in handling diverse image formats within the context of a user-friendly web application.

Aryan Patel:

Undertaking a project focused on Deep Learning in DICOM images presented a steep learning curve, marked by challenges and invaluable lessons. One major hurdle stemmed from a lack of prior knowledge in key technologies such as Python, TensorFlow, and deep learning methodologies. Navigating through these unfamiliar territories demanded dedicated efforts to acquire the necessary skills and understand the intricacies of the tools involved. Another significant issue arose during the implementation phase, specifically in getting the model to run effectively. Debugging and optimizing the model's performance required a combination of patience and persistent problem-solving. Additionally, the project encountered challenges related to data preprocessing and ensuring the compatibility of DICOM images with the chosen deep learning architecture. Each obstacle served as a valuable lesson, emphasizing the importance of continuous learning, adaptability, and a collaborative mindset in the dynamic field of deep learning. Despite the setbacks, the project provided a platform for skill development and a deeper understanding of the complexities inherent in working with medical image data and deep learning techniques.

Leo Kuo:

During this semester, I had the opportunity to dive into this challenging project. A significant part of my role involved rendering DICOM images, designing user-friendly and intuitive front-end pages of our application. As a member of the team, I faced several significant hurdles that greatly contributed to my professional growth. A prominent issue was rendering DICOM images, a complex task due to its unique format in medical imaging. This problem required me to do a lot of research and testing. Furthermore, integrating the U-net model was another steep learning curve. These neural network models were built using Python and Tensorflow, technologies that were relatively new to me. I spent around two weeks learning and digesting python and tensorflow tutorials I found on the internet. This task pushed me to expand my technical knowledge and explore new dimensions in image processing. Working closely with my partner Justin, This project was not only a test of our technical abilities but also a lesson in resilience and adaptability. The semester-long project was an important part of my college journey, teaching me invaluable lessons in building real-world application and problem solving skills.

Paul Ko: Throughout the course of this project, I faced several challenges that demanded a steep learning curve, particularly in dealing with TensorFlow and Python, which were relatively new to our team. The initial difficulty in comprehending concepts such as chunking and voxel distributions, that my partner had implemented, proved to be a challenging task when adding new contributions to the code base. Overcoming these challenges involved leveraging online resources, collaborative problem-solving sessions with my team, and seeking guidance from experienced colleagues. Data preprocessing emerged as a significant hurdle, especially in ensuring the compatibility of DICOM images with our chosen deep learning architecture. To tackle this, I delved into the nuances of DICOM data and adapted preprocessing techniques to align with the model requirements. One major concept from my previous courses that played a crucial role in this project was the understanding of data structures from (CS 3345). This was vital for efficiently handling large datasets and optimizing code performance. One major takeaway from this project was that I was able to learn more about python and AI. I think one way to improve the course is to have mandatory in-person meetings at least once a week with the sponsor to increase engagement.

Sujay Karanam:

Throughout the semester while working on this project there have been countless issues that have hindered my work. One of the first issues was that of the DICOM file and how to get them to the correct shape for the input of the model. After some research, I found a technique called chunking and implemented that into the code. Later on, it would turn out that was not the solution and I went back and made my function that correctly got the DICOM files into the input format needed for the model. Another issue I had was understanding the calculations that were taken to normalize and truncate the voxel distributions. I had to read the paper multiple times to understand the calculations and reasoning behind it to correctly implement the calculations in Python. One of the main issues I had was during the post-processing stage due to the complex steps that were needed to get to the correct output format. I had to do a lot of trial and error to get that part to work so the output could be seen in the desired format since the model's output was just an array of probabilities, the steps that were taken to get that into output format made the array into a probability map and apply a threshold to create binary masks, but to figure out how to implement that in Python was really tough due to the very limited knowledge on the steps required to do that on the internet. A lot of time went into researching these steps. Finally, I am still working on a post-processing issue of making the output more correct by using test cases to evaluate the differences between the old code and the new one I am building. One major thing I have learned during this semester is how to efficiently use Python and TensorFlow for Deep Learning and AI. Before I did know Python and some TensorFlow but not to the extent of what I know now by implementing the model and all the processing that comes with it. One class that I took that really helped me during this semester-long project was the Artificial Intelligence course (CS 4365). It helped by familiarizing me with certain techniques and foundations that I was able to use to write the code such as model training and evaluation, encoding, and probabilities. Lastly, one way that the senior design course can be improved is to have more in-person meetings where all the sponsors and students come in to have talks and more engagement.

FUTURE WORK:

Web Application: Looking ahead, we plan to introduce a more user friendly and appealing look to the page as well as upgrade the program to do multiple files at once but will auto sort and categorize the files uploaded for multiple different patients at once into a server for data. Our commitment is to continuous improvement and innovation in healthcare technology.

GUI: Our GUI has alot of potential to be improved. It can be used in other fields of medicine too that deal with medical imaging. I would like for the GUI to be more flexible in the future with different programming languages, as I believe other programming languages can implement Deep Learning very well too.

FLYWHEEL UI: If our team had some more time to work on the FLYWHEEL UI and the model itself, we would have made some more improvements to the UI and the model. For the model, we would have gone back and improved the processing sections, especially the post-processing part concerning the slices and how they were formatted after getting outputted and reshaped into the correct format for the output screen. We would have reworked how the slices themselves were processed by adding more techniques such as eroding, smoothing, and filling.

SIGNATURES, NAMES, AND DATE

Justin Jung 12/7/23 Justin Jung	Aryan Patel 12/7/23 Aryan Patel

Subhayan Basu 12/7/23 Subhayan Basu	Leo Kuo 12/7/23 Leo Kuo
Sujay Karanam 12/7/23 Sujay Karanam	Paul Ko 12/7/23 Paul Ko
Company Mentor: Dr. Baowei Fei	Faculty Advisor: Dr. Baowei Fei