**AI System Project Proposal**

Project Title:

“**FOCUS: F**ully **O**ptimized **C**onvolutional **U**Net **S**egmentationforTumorDetection”

Project Overview:

The primary goal of this project is to create an AI system capable of detecting tumors using CT scans. This system aims to solve the problem of reducing the costly amount of time that is needed to sift through CT scans images. The expected outcome of this project is to help medical experts identify problems with their patients at a much faster rate leading to earlier intervention, better patient outcomes, and reduced healthcare costs. FOCUS aims to expedite the tumor detection process, enabling medical professionals to make quicker, more informed decisions and ultimately improve patient care. This tool is meant to serve as a diagnostic aid and not a total replacement to medical professionals.

The scope of this project will just include CT scans of the patient's upper abdomen. This project will be the starting point to other potential use cases down the line, but for now only the upper abdomen is being observed due to the simplicity of biological layout and the visibility that CT scans provide. This AI System will be leveraging a convolutional UNet Model to segment tumors formed within the upper abdomen. The data that will be CT scan images of patient’s upper abdomen.

The AI system will employ UNet’s to measure the CT scans in order to segment any potential tumors. The AI system will not train itself to identify if the tumor is benign or malicious because the purpose of this system is to provide experts with the knowledge of any potential tumors that they can test further. Some libraries that will be necessary are PyTorch, Numpy, Pandas, and Matplotlib.

Possible Stakeholders:

* Medical Experts and Radiologists - They benefit from this system by being able to detect tumors much faster than normally
* Patients - They will rely on this system to accurately tell them that they have a tumor in their chest
* Software Developers - They are responsible for creating the software environment to house the AI model
* Data Scientist and AI Engineers - They are responsible for preprocessing the data as well as creating the AI model itself

Possible End Users:

The primary users of this system will be patients that do not know that they have tumors in their chest. They will have minimal interaction with the software itself because it will be the medical experts that process the CT scan through the system in order to bring about a result for the patient.

Possible Other Stakeholders:

Other stakeholders could be the cloud hosting service that might be needed in order to access the AI system. There’s also the hospitals and clinics that might use the software as well.

**Computer Infrastructure**

Project Needs:

The primary objective of FOCUS is to segment tumors within medical images of patients such as CT scans which means that my project will take use image data to train itself on. Some basic performance benchmarks I have set for FOCUS are accuracy, latency, and throughput. I plan on prioritizing these metrics in the following order. The reason I value accuracy over any of the other benchmarks is because the dangers of outputting too many false negatives is risky. Details will be explained later in the project proposal.

Ideally, FOCUS would be hosted on a cloud-based server so that clinics and hospitals with internet access may easily use FOCUS whenever they please. The constraint with doing it this way is that it is costly to maintain a cloud-based webserver. I believe this is an acceptable constraint to have since this type of tool would need some high-performance computing done by GPUs in order to make inferences using the model. Since hospitals are not known for having state of the art technology, it just makes sense to offload the heavy computing to a cloud server. This would also mean that this tool would not be accessible under the circumstances of no internet connection.

Hardware Requirements:

Some of the hardware requirements I would need for training my model would be a single GPU such as the A100 that is provided by the school. The plan is to use the A100 on HiPerGator to do most of the training and inferencing for FOCUS. From my experience with working with 3-dimensional tensors like CT scans, I will probably need at least 30 GB of memory, an A100 GPU, and 30 GB of storage. Having 30GB of memory will be important because I am working with 3D images. 3-dimensional tensors take up a lot of space. As for why I would need an A100 GPU, this would allow me to expediate both training and inference time for the system. For the 30GB of storage, this will be important in housing the dataset itself as well as the model. The images alone take up more than 2/3 of that storage space itself.

Software Environment Planning:

I plan to use Windows as my OS due to the comfortability I have using it. As for my software stack I plan to use the following:

* AI Frameworks
  + PyTorch
* Libraries
  + NumPy
  + Nibabel
  + Matplotlib
  + Monai
* Cloud Container
  + Docker

The reason I plan to use Monai and Nibabel is because they are useful libraries when it comes to loading medical images such as .nii.gz files.

Cloud Resources Planning:

Although development will be done on HiPerGator, when it comes to deployment the idea is to use Docker for hosting the system. Docker is convenient in being able to compute, auto-scale and store any related data to the system.

Scalability and Performance Planning:

Some performance optimization techniques that are in consideration for my AI system are post-training quantization, trying to make the model more lightweight and decrease latency. There is also pruning, which cuts off unimportant weights from the neural network to improve inference speed and decrease latency. I was also considering using augmenting the training data as well which I believe is especially helpful due to the small sample of images compared to many other deep learning models training data.

Some strategies I plan to use for scaling resources based on workload demand is to use Nsight Systems. This is a service that helps keep track of GPU activity as well as trace GPU workloads. This will be helpful to monitor the performance of my system and increase active GPU usage during high workloads or vice versa.

**Security, Privacy, and Ethics (Trustworthiness)**

Problem Definition:

**Goals –** Protect confidential user data as well as ensure the model is fair for all users so that users trust their data with our system.

**Strategies –** Develop an ethical impact assessment to evaluate potential risks related to data leaks. I could also develop a risk analysis framework to help organize potential risks related to the model training. This would ensure help ensure that the model is being trained fairly and let me figure out what to prioritize and what to focus on. These strategies could potentially be aided by the use of a framework called the Data Ethics Canvas (ODI) which is a tool that would help me use data ethically.

Data Collection:

**Goals –** Collect data through ethical means such as utilizing public CT scans for training and implementing data augmentation on these images.

**Strategies –** In order to collect my public data, I will be using the public data provided by my research advisor. I also plan to use Kaggle to supplement these CT scans if I am in need of more data for training. As for data augmentation I only plan to use nibabel to not only load my data into the model. For the transformation itself I will be using Monai. Monai is a python library specifically made for medical imaging, so I believe it will be important for my use case.

Model Development:

**Goals –** Developing a fair and robust AI model that will give accurate and quick tumor segmentations of the FOCUS.

**Strategies –** For my strategy to build a robust AI model, giving the model different sized CT scans will make the model much more robust. The model will also take care of certain preprocessing steps to make sure that different sized CT scans get resized to a more uniform distribution.

Deployment:

**Goals –** Deploy FOCUS on the cloud and ensure that it performs well in real world conditions.

**Strategies –** My strategy to maintain a strong model would be to implement consistent model updates periodically every couple months to ensure that the model’s performance is not degrading. This would allow my model to continuously update and rollback in the case of an emergency. For my use case, I would use BentoML to create an environment that allows for me to continuously monitor the system over production.

Monitoring/Maintenance:

**Goals –** Continuously monitor model performance by making periodic model evaluations to FOCUS in order to maintain user trust.

**Strategies –** This strategy can be implemented as I mention previously using BentoML to create an environment that would more easily allow me to keep track of my model’s performance. Implementing BentoML would allow me to detect any anomalies within the model’s results and allow me to revert back to previous versions of healthier models.

**Human-Computer Interaction**

1. **Defining HCI Requirements During Problem Statement and Requirements Gathering**

Understanding User Requirement:

For my project I believe that utilizing google forms will be a viable method of gaining insight into the problems that medical practitioners may face using the current software they have for medical imaging. This is because it is important that I understand the current technology that these practitioners are using to discover tumors. It would also be a good idea to interview some of these practitioners to get an even better understanding of their workflow by asking questions during the interview that may arise from speaking with them. Something that google forms would be limited to. The general outline of the form would try to highlight specific pain points or inefficiencies that healthcare workers face when dealing with identifying tumors on their own.

Creating personas and scenarios:

So, for my persona I believe only a single persona is enough for my project. There should only be expert users that deal with my tool. Although this tool is made to aid both medical practitioners and the patients in hospitals, there should be no actual interaction between the system itself and any patients.

Persona 1: Dr. Mike

* Background
  + Dr. Mike is a radiologist working at Shands Hospital. One of the aspects of his job is to look over CT scans that other doctors provide to him to check for tumors. He has been doing this for a long time however due to the stress that the American healthcare system puts onto their workers, it appears that the rate of which Dr. Mike takes to look over these CT scans is not fast enough. Shands would like to hire another professional to assist Dr. Mike with his job, however the funds are not there. Now Dr. Mike is in a pickle because he has no way of performing this task any faster than he does now.

Conducting task analysis:

**A screenshot of a computer screen

Description automatically generated**

This is the approximate layout in which the user will be interacting with the system. There will be a CT Scan upload box which will send the images to the backend of the system where that data will be processed and output the CT scans with the segmentation of any potential tumors. To put this into a list of steps for a beginner user:

1. Read the description box on how to use the tool
2. Prepare the CT scans to be uploaded into the system
3. Wait for processing of the images
4. Look over the Segmentation of the CT scans to see if the images show any tumors within the patient
5. Look at the likelihood of a tumor appearing in the CT scans and judge as a practitioner whether these scans are worth taking a second look at

Some potential challenges that a user might face are utilizing the likelihood of tumor detection. Since this system will be using deep learning techniques to determine if a tumor is present within CT scans, it will have to be up to the practitioner to thoroughly investigate the images themselves to see if the system was correct.

Identifying accessibility requirements:

Due to the inherent nature of computer vision tools, it might be difficult to meet accessibility standards, however in order to do so I would make sure that the front-end interface of the system does not have a lot of dynamic components to the software. The reason why I think keeping this in mind when designing software is because screen readers have a lot of issues when dealing with dynamic moving websites. Also making sure that text boxes are clearly labeled to ensure that visually impaired users can utilize my system will also be important.

Outlining usability goals:

Usability Goals:

1. Maximizing accuracy of model
2. Minimizing segmentation task completion time
3. Hearing positive feedback from medical practitioners about the software

For the first 2 goals, I plan to use TensorBoard to measure the accuracy of completion time of the system. This is a free service provided by PyTorch that is capable of measuring the evaluation metrics of a PyTorch neural network. As for my last goal, I plan on doing semi-structured interviews of actual medical practitioners utilizing the software that way I am able to hear about the positives and negatives of my system.

1. **Apply HCI Principles in AI Model Development**

Develop Interactive Prototypes:

**A screenshot of a computer screen

Description automatically generated**

Using the prototype previously, this is how the design of my AI system will look. As you can see there will be a box dedicated to explaining how FOCUS works which will help beginner users with first time use. There will also be an upload button to upload the CT scans as well as an output box for the segmented image. Finally, there will be a box dedicated to showing the likelihood of a successful segmentation.

Design Transparent Interfaces

I plan to use python libraries such as matplotlib to visualize the segmentation masks created by the UNet model. This will be useful in understanding the results produced.

Create Feedback Mechanisms

To incorporate user feedback, the plan will be to provide a link that will lead to survey about the user experience with FOCUS. Using this experience, the future of FOCUS can be developed towards making users more satisfied.

1. **Prototype and Test User Interfaces During System Design**
2. **Ongoing Monitoring and Iteration Post-Launch**

**Risk Management Strategy**

Problem Definition

Some key risk considerations that are being taken into consideration for this stage are certain liability risks that naturally come up when medical practitioners rely on technology. For example, if FOCUS had misidentified a tumor cell as an ordinary part of the body, this would potentially make the proprietor of FOCUS and the medical practitioners liable for missing something that is clearly harmful to the patient. The strategy to avoid situations like these from happening is for the model to be trained to skewered towards improving recall which would make missing positive observations any tumors as small as possible.

Data Collection

An important risk that needs to be addressed for FOCUS is the issue of data privacy. FOCUS will need to be trained on real data from patients with tumors. Because this is the case, it is of most extreme importance that no patient’s personal information should be linked to the model that could potentially tie back to the patient. For this stage, the plan to prevent this from happening will be to use publicly available datasets that have personal information wiped off already.

Model Development

During this stage there is a risk of overfitting the model to the training data due to the relatively small sample size of CT scans available. The plan to mitigate this risk will be to apply data augmentation techniques to the existing training dataset such as random spatial crop to make the data more generalizable. During training, an image from the training set will be pulled and used as validation image every epoch, that way the model also for the purpose of increasing generalization.

AI Deployment

Some problems that may occur during this stage is the risk of integration issues with current medical practitioner workflows. New technology such as this one may be difficult for some practitioners to use at first which is why to mitigate this risk, I will be interviewing with practitioners to inquire about the exact process in which they go through in order to determine if a patient has a tumor using CT scans. Hopefully, these interviews will glean whether FOCUS will be able to easily integrate into current workflows.

Monitoring and Maintenance

Just like many AI systems, FOCUS is susceptible to a risk of performance degradation over time. To mitigate this from affecting practitioners and patients alike, biannual inspections of the model’ performance can be done onto the system in order to measure if the effectiveness of FOCUS is still strong or may need maintenance.

Residual Risk Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Impact** | | | | |
| **Likelihood** |  | 0  Acceptable | 1  Tolerable | 2  Unacceptable | 3  Intolerable |
| Improbable | Integration of FOCUS into workflow |  |  | Data Privacy  Liability Risk |
| Possible |  | Performance Degradation over Time | Overfitting on Training Data |  |
| Probable |  |  |  |  |

**Data Collection Management and Report**

Data Type

The type of data that FOCUS will be using is unstructured because this project will be using CT scans of patient’s bodies with potential tumors. Some challenges that come with handling this sort of data is the type of preprocessing that is necessary in order for the model to handle the data. One such preprocessing step being resizing the images to be consistent for all the data.

Data Collection Methods

The source of my dataset was attained through my research lab, Gong Laboratories. There are no problems with the reliability of the source of data and all the data appears to be annotated correctly. The data was downloaded from a repository associated with the lab. As for ingestion there are two parts for this: training and deployment. For training, the images were loaded into batches in order to train the most amount of data as efficiently as possible. For deployment the data is being ingested in real time in order to get the results for tumor segmentation to the practitioner as soon as possible.

Compliance with Legal Frameworks

The use of AI in healthcare, especially for medical imaging, must be approved by the U.S. Food and Drug Administration (FDA). As of March 2024, the FDA published the “Artificial Intelligence and Medical Products: How CBER, CDER, CDRH, and OCP are Working Together” which represents a coordinated approach to AI. Although the FDA is working towards regularization of AI, it is still a fast-paced field and the legal frameworks around AI are not fully in place yet.

Data Ownership

The data is owned by the University of Florida because it is stored on HiPerGator. To access this data, a two-factor authentication process must be submitted in order to log into HiPerGator and access the data. It is important to have these types of access controls because data privacy is a prevalent issue for all AI systems. Especially for a system such as FOCUS that handles personal patient health data.

Metadata

With NifTi images, these come with an associated header of data such as dimensions, attributes related to each slice of data as well as the size of the image. There has been no issue encountered while using metadata because there has not been a need to use the meta data for these images.

Versioning

For versioning, FOCUS will be using Git in order to keep track of the different versions of data that this process may partition through. That way the data cannot be lost or damaged in anyway without the possibility of recovering the data.

Data preprocessing, Augmentation, and Synthesis

For preprocessing, the only technique that was used was resizing in order to fit all the images into FOCUS. Preprocessing was not a big issue with FOCUS because the data provided was already annotated well. The only issue was the issue of different sizes among the training data.

This project was more heavily focused on data augmentation and coming up with creative solutions to make do with the limited amount of data available for training a model. The main technique that was used to augment the data for training was to use a technique called random spatial crop which essentially took a subset of the image and used that subset as a training image for the model. Another augmentation technique that was considered was flipping the images in order to double the size of the data.

Report on Risk Management in Data Collection

An important risk that needs to be addressed for FOCUS is the issue of data privacy. FOCUS will need to be trained on real data from patients with tumors. Because this is the case, it is of most extreme importance that no patient’s personal information should be linked to the model that could potentially tie back to the patient. For this stage, the plan to prevent this from happening will be to use publicly available datasets that have personal information wiped off already.

Report on Trustworthiness in Data Collection

The goal is to collect data through ethical means such as utilizing public CT scans for training and implementing data augmentation on these images.In order to collect my public data, I will be using the public data provided by my research advisor. I also plan to use Kaggle to supplement these CT scans if I am in need of more data for training. As for data augmentation I plan to use Monai to not only load my data into the model, but also apply any relevant transformations to my model. Monai is a python library specifically made for medical imaging, so I believe it will be important for my use case.

**Model Development and Evaluation**

1. **Model Development**

Algorithm Selection

The model that I am considering for my project will be a convolutional UNet model because this deep learning architecture has some significant results in real-world segmentation, and I believe that this model will perform well on segmenting tumors in medical images. The reason I think this is because a big problem with medical images is that they need to have a lot of image variability and noisiness. The benefit that I believe this UNet model will bring is that it should be able to use convolutions to produce an accurate segmentation, therefore saving a lot of time for medical experts.

Feature Engineering and Selection

For feature selection, I created a function that shrinks the image to mainly encompass the areas that contain the tumor. The reason for this is because the random spatial crop size I am using (96, 96, 96) is randomly cropping into unsegmented space too often hence the need to shrink the medical images.

Model Complexity and Architecture

For my model architecture, I will address the multiple components that make up the UNet. There are 2 major components that need to be addressed for a UNet: the up sampling and down sampling blocks. To prevent overfitting, I will be using cross-validation strategies common in medical imaging such as leave-one-out in order to make the model better at generalization.

1. **Model Training**

Training Process

In terms of my training process, using mini-batches as small as 8 images per batch were considered because the amount of training data available is much smaller than normal deep learning training datasets. The optimizer for this project will be using Adam which is a fairly common optimizer used in deep learning. To measure the performance of mask segmentation, this project will be using Mean Dice. The reason is because Mean Dice is a metric that compares the similarity of images which will be helpful when comparing the ability of the model to segment data properly. For the learning rate, the best result so far is 1e-3, but that experimentation is still proceeding. As for the number of epochs, the initial stage of training has the number of epochs at 3 to 5. This is because training is very computationally expensive, so the priority is to find the best hyperparameters for the other values first before continuing with a greater number of epochs.

Hyperparameter Tuning

The key hyperparameters for this experiment are learning rate, number of epochs, and batch size. GridSearchCV may be a viable technique towards tuning these hyperparameters. Results from the training are still pending. The method of monitoring that was chosen for this project was to record the loss and mean dice score of each training session. This data was then displayed on graphs using TensorBoard in order to monitor for underfitting and overfitting.

1. **Model Evaluation**

Performance Metrics

The metric chosen for this project is Mean Dice. The reason why this was picked is because it is a common metric used to compare image segmentations. A Dice score is generated by multiplying the number of common elements in a segmentation by 2 and then dividing that number by the total number of elements in both images. Since there are multiple images, the Mean Dice score is just the average of all of those Dice scores. This makes this metric perfect for my project since I will be needing to compare the different image segmentations created by the UNet. As for providing the model’s performance metrics, those will be attached soon.

Cross-Validation

For cross-validation, this project uses the leave-one-out strategy which is common in medical imaging. The reasoning behind this CV strategy is because of the limited amount of training data provided. The leave-one-out strategy, although not ideal, is effective because it takes into consideration of how small the training dataset is and therefore would help improve generalizability. Results from cross-validation will be provided later.

1. **Implementing Trustworthiness and Risk Management in Model Development**

Risk Management Report

The risk that was identified for model development specifically was the model overfitting on the data due to the small training dataset. This however, has not been an issue as of yet due to the fact that the model has actually underfit on the data so far. The reason behind this is still being looked into at the moment so there is not much to say about the issue of overfitting. As for the risk of underfitting, this has most definitely come into play. The strategy I plan to tackle this problem will be to experiment with different hyperparameters and preprocessing techniques to try and understand why the model is underfitting.

Trustworthiness Report

One of the goals for implementing trustworthiness in the model development stage is to try and implement FoolBox which is a tool that will try attack the model with malicious input data. Because the model has not been fully trained yet there has been no results on implementing this just yet, but once the model is finished training this is one of those corresponding steps.

1. **Apply HCI Principles in AI Model Development**

Develop Interactive Prototypes

For this project, the plan is to use Streamlit create a interface to for user to interact with the model. The goal is to include a text box describing how the system works, an input box to upload the medical data, a output box that will spit out the image with the segmentation, as well as the probability of confidence that the model is correct. Here is what the initial outlook of the interface will look like:

**A screenshot of a computer screen

Description automatically generated**

Design Transparent Interfaces

The plan is to use medical imaging libraries such as nibabel, which are specifically geared towards working with medical imaging data, to display the images.

Create Feedback Mechanisms

To incorporate user feedback into the interface, there could be a button that leads to a google survey that would ask questions such as:

* How did you enjoy using FOCUS?
* How could your experience be improved?
* Did you find FOCUS to be a useful product?

**Deployment and Testing Management Plan**

Deployment Environment Selection

Deployment will be done through the cloud using a combination of streamlit and docker. The reason why I choose to have a cloud-based platform is because in production many hospitals will want to be able to access this kind of technology. However, due to hardware restraints doing so locally would not be feasible for many hospital/clinics considering you would need a GPU of some sort to do the inferencing. Doing so on a CPU would affect inference latency which would prove detrimental to the fast-pace lives of the doctors.

Deployment Strategy

I plan to use docker to put my project into a container. I believe that Docker is the right tool for me because this fits the reliability and scalability goals that I have in mind for FOCUS. What is great about using Docker is that it is lightweight meaning scalability would not be a problem. What is also good about Docker is that because of the way Docker is structured, it is not easy for hackers to break into my container. This is an important concern for my project considering I am handling sensitive patient data. Docker is also transferable among different servers which means that I don’t have to worry about the different types of operating systems that hospitals may be using for themselves. This is what makes Docker both reliable and scalable.

Security and Compliance in Deployment (Trustworthiness and Risk Management)

So referring back to the Trustworthiness and Risk Management sections, Docker actually solves the issue of making sure that patient data is reliably stored safely. As I mention previously, the way Docker is structured, it is not easy for hackers to break into a container because a container only has the minimal number of ports used which limit hackers access to the information within the container.

**Evaluation, Monitoring and Maintenance Plan**

System Evaluation and Monitoring

For performance monitoring I plan to use a simple logging system supplemented by Streamlit’s inhouse capabilities already. As for the evaluation metric I will used I believe that using Mean Dice will still be an effective metric of comparison. This is because I will be able to measure the model’s current performance compared to previous results. If I were to see a continuous dip in the Mean Dice, that would be a good indication that the model is not performing as it should be. This is especially the case if the model were to see CT scans that are very different from the training data itself. Following a dip in Mean Dice, the method of fixing this would be to further increase the training data used by FOCUS in order to improve on its generalizability.

Feedback Collection and Continuous Improvement

I plan to implement a simple feedback mechanism using Streamlit. There will be a button that will take in user feedback and store it within an Azure database. This feedback form will list two questions. Whether the user thought the product would be useful for increasing efficiency in their workplace and if the user wanted to leave any written comments about FOCUS.

Maintenance and Compliance Audits

For maintenance of FOCUS, the updates will most likely be periodic updates. Once every month, the system will pull the old training data as well as the new training data collected from production to be used to ensure that the model performs well. This would address the potential risk of overfitting. Also to ensure trustworthiness of the data, there will be a generated set of fake CT scans that will be tested in model to ensure that the model is not approving of fake images and the like. This is how I will evaluate the trustworthiness of the system. In the case of a good result from bad images, the model will have to train again off of a different subset of training images until the model is deemed appropriate to use. This standard will most likely be set by the initial Mean Dice reproduced after initial training.