The Potential of Artificial Intelligence to Identify Cancer and Aid Research

Author(s) Wilson B. Peguero Rosario

Master of Science in Data Science Capstone Project Proposal

Grand Canyon University

Instructor: Professor TBA

Revision: 5

Date: November 9, 2022

**ABSTRACT**

Artificial Intelligence (AI), the science and engineering of creating intelligent computer programs that can think rationally and without human interaction. This field has rapidly expanded within the last five years, causing massive changes within all industries, from Medicinal, civil, finance, to even government agencies that currently leverage AI to minimize long and arduous tasks to simple calculations.  These AI models can think on a much faster level than the human mind can process and can make assumptions from existing data and even predict what may occur in the future. Because of these capabilities, AI is one of the most invested skills within the field of medicine. With the ability to classify certain images based on different stages of cancer, a doctor may be able to pick out early signs that can save a patient’s life. With the ability to predict the reaction between chemical combinations and the human body, AI can assist pharmaceuticals in non-destructive drug discovery. The potential of AI, specifically in cancer research, cannot be ignored any longer.

**PROJECT SYNOPSIS**

The capstone project is based on the ImageNet challenge in conjunction with the desire to develop a machine learning model that can accurately diagnose tumors as either malignant or benign based on an image and some patient information. The VGG-16 algorithm (explained later) was modified to include patient metadata as input parameters and used to train the machine learning models that are used to then make predictions. The goal of the project is to develop model that performs with an accuracy of at least 90% and assists radiologists in diagnosing tumors based on images. The machine learning model’s best performance has been 71%, this may be attributed to the lack of data, which does not allow the model to appropriately classify as a tumor either as benign or malignant. Alternatively, the machine learning model may have better performance if allowed to focus on the area of interest. Given more time the model may improve in terms of performance with better fine-tuning.

|  |
| --- |
| **HISTORY AND SIGN-OFF SHEET** |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 5/30/2022 | Wilson Peguero Rosario | Project Overview and Abstract for milestone 1 |
| 6/11/2022 | Wilson Peguero Rosario | Initial draft for review/discussion for milestone 1 |
| 6/22/2022 | Wilson Peguero Rosario | Final Draft for milestone 1 |
| 8/8/2022 | Wilson Peguero Rosario | Initial draft for review/discussion for milestone 2 |
| 8/31/2022 | Wilson Peguero Rosario | Finished Draft for Submision of milestone 2 |
| 11/3/2022 | Wilson Peguero Rosario | Updated objectives to represent DICOM files for milestone 1 |
| 11/9/2022 | Wilson Peguero Rosario | For milestone 1:   * Updated the project scope to use image data AND header file data * Included assumption about header files * Included overfitting and sample size to risk table * Reworked the schedule hours and included web design in work breakdown table |
| 11/12/2022 | Wilson Peguero Rosario | For milestone 2:   * Finished Design Planning summary by setting back the scope to only diagnose tumors based on malignancy. * Update description to include One Hot encoding of labels rather than ordinal encoding * Changed GPU hardware from 2060 to 3090TI |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

*<Insert name of instructor here>*

**TABLE OF CONTENTS**

milestone 1

Project Overview and Project Objectives 5

Project Scope 6

Project Success Measures 7

Project High-Level Solution 8

Project Controls 9

Project Cost and Schedule 10

Appendix A – References 11

Appendix B – Copyright Compliance 14

MIlestone 2

Design Planning summary 15

Overview of model pipeline design 16

Detailed model pipeline design 17

Appendix A – Technical Issue and Risk Log 22

Appendix B – External Resources 23

MIlestone 3

System Entity 24

Functional Requirements 24

source code listing 25

Code Review 30

Implementation Plan 30

Application Functionality and Execution 31

User guide 32

appendix a - references 35

milestone 4

components testing 36

Requirements testing 37

system testing 40

Project Overview and Project Objectives

**State the Problem and Background**

Cancer has existed since the ancient Egyptians (Husaini et al., 2020). As technology advanced over the eras, so did our knowledge of the potential cancer have as a disease. Multiple ways of identifying cancer have been utilized over time. These modalities are called Mammogram, Ultrasound, CT scan, and Thermography (Devita et al., 2016). These imaging modalities can be supplemented with machine learning algorithms to detect early signs of cancer (Cardoso et al., 2020), and the correct form of treatment (Hadjiyski, 2020). Multiple cancer treatments have been developed through drug discovery and gene sequencing. Machine learning can supplement cancer treatments through non-destructive drug discovery, which will save resources and materials used to develop the medicine up to the clinical trial stage. Through AI-powered gene sequencing, cancer treatment can be customized on a patient level to provide efficient treatment while avoiding other more harmful treatments (Kulkarni et al., 2019). AI-powered gene sequencing can go as far as working on a genetic level to suppress certain genome sequences that can cause cancer to develop in the patient (Iyer et al., 2019).

**Christian Worldview**

Cancer is the second cause of death in the United States. This is a painful experience both for the patient and the family that can be traumatizing for those who are misdiagnosed and for those who do not require special treatment as the tumors are not malignant but rather benign. As a Christian, I am meant to associate and empathize with the pain of my fellow man or woman. To attempt to help them in any way that I can is something that God has called us to do. As someone who had a family member to die from breast cancer, I find that largest factor to cause this family members death was not due to the doctors, but rather the limitations that they have from having to handle multiple people suffering from different ailments and proper customized treatment. I can’t bring back my family member back to life, but I can help others from suffering the same thing that my family did.

**Project Objectives**

* Gather data set of DICOM files with images that are labeled as either benign or malignant (these images will contain tumors).
  + The DICOM file will contain textual, numerical, and categorical data within the header file and image data.
* Develop a base model to use as a baseline for comparison.
* Develop model class with potential to extract features from DICOM files.
  + Learn ideal features to extract from DICOM file.
* Train model based on gathered data and use some of the gather data as a test set.

**Challenges**

Data Gathering will be the biggest challenge. Although there are multiple medical image data sets, majority of the data sets are not reliable enough to surpass current misdiagnosis rate (95%). There is also the case that the images are using different modalities (i.e., X-ray, MRI, Ultrasound, etc.).

**Benefits and Opportunities**

This project may be utilized to start a biomedical software company or to use as project for hire in a medical institution. The gap between the images taken for each patient and the number of radiologists available to diagnose patients makes this market very profitable and easy to introduce, such that there are currently multiple institutions researching and investigating how the gap may be filled using machine learning algorithms to assist radiologists in diagnosing cancer through medical images.

Project Scope

Develop a machine learning model that can classify patients on whether a tumor is malignant or benign using DICOM file data with the same accuracy as the annual misdiagnosis rate.

In Scope Features:

* Model with ability to classify based on medical image data AND patient metadata.
* Able to receive raw image, and patient metadata to classify the patient.

Out of Scope Features:

* Highlights regions of interest.
* Generates details regarding the image.

**Table <Insert #>. Stakeholders**

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| Self | Data Scientist | Gathering data; developing model; training model; test model; interpret results |
| Operator | Data Analyst | Developing the model; Training and testing the model |
| User | Radiologist | Interpreting the results |
| Maintainer | Data Engineer | Gathering the data |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Collect Data | Data availability | Finished | 40 | Free |  | Fully Labeled Data Set | 7 Days | 9 days | Kaggle; UCI Machine Learning Repository; GitHub |
| 2 | Process Data | Data Availability | Finished | 20 | Free |  | Fully Filtered Data Set | 3 Days | 5 Days | Data Collected |
| 3 | Research Machine Learning Models | Journals with model designs | Finished | 40 | Free |  |  | 7 Days | 3 Days | Journal Articles |
| 4 | Design Model | Libraries available;  Programming languages;  Data type;  Research articles | Finished | 100 | Free |  | Completed Model Design | 15 Days | 5 Days | TensorFlow;  Python;  NIH; |
| 5 | Training & optimize hyperparameters | Libraries available;  Research Articles; Model Design | In Progress | 80 | Free |  | Optimized Model | 14 Days |  | TensorFlow;  Python;  NIH |
| 6 | Test Model | Data Collected | In Progress | 10 | Free |  | Fully Tested Model | 2 |  | TensorFlow;  Python;  NIH |
| 7 | Evaluate Model | Data Availability; Research Articles | In Progress | 20 | Free |  | Model with real life evaluation | 4 Days |  | TensorFlow;  Python |
| 8 | Develop Dashboard | Research articles; Model performance; Libraries available | Finished | 8 | Free |  |  | 1 Day | 5 Hours | Python; Dash library |

Project Success Measures

The metrics used to measure project success will be based on the model’s ability to label the medical images as either malignant or benign. The loss, accuracy, recall, and precision metrics will be used to compare with how well the model performs in the diagnosis aspect of this project. Other metrics, such as AUC will be used to visually explain the performance of the machine learning model.

|  |
| --- |
| Project Completion Criteria |
| 1. Accuracy matches the range provided within articles containing similar models (60%-90%) |
| 2. Accuracy of the model is equivalent or higher than the yearly correct diagnosis rate (95%) |
| 3. false positive rate of the model is minimal (5%-10%) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | The data sets from multiple hospitals are not significantly different from each other in terms of imaging modality | There are some parameters used to take MRI images and even different devices to take images of the tumors which may impact the model’s ability to classify tumors. There are also changes in resolutions related to the image which may also impact the model efficiency, these are assumed to have minimal impact. | Assumption |  | 6/11/2022 |
| 2 | The data sets will stem from at most 20 different hospitals |  | Constraint |  | 6/22/2022 |
| 3 | The images will be taken using at most three (3) imaging modalities (MRI, Computed Tomography, SPECT) | The three main imaging modalities used are Magnetic Resonance (MRI), X-rays (Computed Tomography), and Nuclear Medicine (SPECT) | Constraint |  | 6/22/2022 |
| 4 | All available header file data pertaining to the patient correlates with the malignancy of the tumor | Most of the information gathered regarding the patient pertains to risk factors of cancer. | Assumption |  | 11/5/2022 |

Project High-Level Solution

**Introduction**

Current Cancer treatment has reached its limits. Cancer screening is very effective at discovering cancer in its early stages, yet the mortality rate has not significantly changed because of increased cancer screening (Devita et al., 2016). The efficiency of early screening regarding mortality is heavily dependent on the treatments available for the types of cancer (Devita et al., 2016). There are too many factors leading to cancer that can be observed in a typical lab environment. Tobacco alone has enough of an effect to be considered as the cause of cancers located in the bladder, cervix, colon, and rectum, esophagus, kidney larynx, leukemia, liver, lung, oral cavity, and pharynx, pancreas, and stomach (Devita et al., 2016). Although current cancer treatments are limited, catching cancer at its earliest stages yields high survival rates (Kulkarni et al., 2019).  These factors cause the survival rate of cancer to stagnate at very low percentages, as cancer is not caught on time, as more cancerous devices spread, such as the popularity of the e-cigarette, or vaping, so do the chances of developing cancer. The highest probability of survival occurs in the earlier stages (Devita et al., 2016), creating a need for AI to better identify cancer.

**Solution**

The project can be split into three sections, data gathering/mining, data validation/analysis, and AI modeling. The first step in the procedure is to gather imaging data as well as medical test results from multiple sources (such as Kaggle, IEEE, government websites, associate organizations) to create a valid dataset that can be leveraged to prove the potential in AI through testing and comparisons with raw data. Prelabelled or filled in data would be required to use comparative analysis and observe the efficiency of the machine learning model to identify or fill in for missing labels or values. Once the data is gathered, the next step is to properly validate the data. This would include removing outliers, finding importance in features, identifying independent and dependent features, and minimize the dataset to its most useful and functional version. Then, the machine learning model will be developed based on research articles and peer-reviewed journals and trained, its hyperparameters will also be optimized to develop the best model possible. This will require majority of the project time as the development of the model will require fine tuning and the most efficient model used may require a combination from multiple algorithms. Lastly, the model will be tested extensively on pre-labeled test data. Comparative analysis is done to see the efficiency of the model against pre-labeled data.

Project Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RISK MANAGEMENT | | | | |
| **Event Risk** | **Risk Probability**  **(high, medium, low)** | **Risk Impact** | **Risk Mitigation** | **Contingency Plan** |
| Data from different image modalities cause inaccuracy | medium | The model will inaccurately label medical images | adding hidden nodes to separate and classify images based on imaging modality. | Use only data sets using only one imaging modality. |
| Random noise | High | Minimal | Can be mitigated by filtering out noise | Remove data from the training set. |
| Overfitting | High | High | Add dropout layer(s) which will turn off the output of | Use smaller learning rate for the optimizer |
| Low Sample Size | Medium | Low | Include different data sets | Data augmentation techniques |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ISSUES LOG | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | Anonymized data limits model design | Will minimize the scope of the project to simple, and separate algorithms | Create separate models to evaluate different metrics or sets of data | Wilson Peguero Rosario | *High* | *6/22/2022* |  |  |
| 2 | Limitations of hardware equipment | Will extend the schedule past its targets. | Fine tune the training process to maximize model accuracy while minimize time spent training | Wilson Peguero Rosario | High | 6/22/2022 |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CHANGE CONTROL LOG | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ROLES AND RESPONSIBILITIES | | | |
| Name | Team | Project Role | Responsibility |
| Radiologist | N/A | Input on GUI | To comment and provide feedback on the model and its results. |
| Data Analyst | N/A | Analyze data | Review the feedback on the models and retrain based on comments |
| Data Engineer | N/A | Maintain Pipelines | Make improvements based on the comments that Data Analyst sends their way in regards to the data. |

Project Cost and Schedule

There will be no project cost as this project can be achieved using already purchased computer hardware. The kind of hardware that one possesses will only impact the time the model takes to be trained and fine-tuned.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Collect Data | Data availability | Complete | 40 | Free | 9/30/2022 | Fully Labeled Data Set | 7 Days | 14 Days | Kaggle; UCI Machine Learning Repository; GitHub |
| 2 | Design Model | Libraries available;  Programming languages;  Data type;  Research articles | Complete | 100 | Free | 10/15/2022 | Completed Model Design | 15 Days | 30 Hours | TensorFlow;  Python;  NIH; |
| 3 | Training & optimize hyperparameters | Libraries available;  Research Articles; Model Design | Complete | 80 | Free | 11/01/2022 | Optimized Model | 14 Days | 2 Days | TensorFlow;  Python;  NIH |
| 4 | Design Dashboard | Libraries available | Complete | 10 | Free | 10/25/2022 | Dashboard | 1 Day | 5 Hours | Python; Dash; Plotly |
| 5 | Evaluate Model | Data Availability; Research Articles | In Progress | 20 | Free | 11/5/2022 | Model with real life evaluation | 4 Days |  | TensorFlow;  Python |

Appendix A – References

Bongiolatti, S., Gonfiotti, A., Borgianni, S., Crisci, R., Curcio, C., Voltolini, L., Alloisio, M., Amore, D., Ampollini, L., Andreetti, C., Argnani, D., Baietto, G., Bandiera, A., Benato, C., Benvenuti, M. R., Bertani, A., Bertolaccini, L., Bortolotti, L., Bottoni, E., & Breda, C. (2021). Post-operative outcomes and quality of life assessment after thoracoscopic lobectomy for Non-small-cell lung cancer in octogenarians: Analysis from a national database. *Surgical Oncology*, *37*, 101530. https://doi.org/10.1016/j.suronc.2021.101530

Data from a database regarding Video-Assisted Thoracic Surgery Lobectomy (VATS-L) for Non-Small-Cell Lung Cancer (NSCLC) was collected key metrics to observe the outcomes for the elderly. The authors were able to derive metrics from the dataset, such as “...30-day and 90-day postoperative mortality...[, ]any complication...[, ]complication rate...” and more to determine whether there is a significant risk that the elderly between 80-89 have when undergoing VATS-L for lung cancer through machine learning. Elsevier first received this article on 22 October 2020, revised on 27 December 2020, and accepted on 25 January 2021. The authors of this article utilize machine learning algorithms (logistic regression) to derive metrics related to complications that did not exist within the dataset, demonstrating the potential that AI possesses to supplement existing data and provide a new narrative on the risks of surgery for a specific age group.

Cardoso, M. J., Houssami, N., Pozzi, G., & Séroussi, B. (2020). Artificial intelligence (AI) in breast cancer care - Leveraging multidisciplinary skills to improve care. *The Breast*, 110–113. https://doi.org/10.1016/j.breast.2020.11.012

This article focuses on the potential of AI to automate repetitive tasks in the medicinal field and allows medical professionals to focus on providing a better patient experience. The article describes the different fields that AI can apply to, such as medical imaging, predictive diagnoses, and more. This article is a part of “The Breast” Journal, which is peer-reviewed by an editorial board comprising medical doctors and doctors from other professions (i.e. biomedical engineering). This article shows the potential that AI has in multiple fields related to medicine and can be utilized as a part of the introduction to AI in the medical field.

Devita, V. T., Lawrence, T. S., & Rosenberg, S. A. (2016). *Cancer : principles & practice of oncology. Prostate and other genitourinary cancers*. Wolters Kluwer. https://gcu-encore.iii.com/iii/encore/record/C\_\_Rb1321879\_\_SCancer%20\_\_P0%2C1\_\_Orightresult?lang=eng&suite=def

The book provides in-depth information about cancer, the medical techniques used to identify the type of cancer within the body, the risk factors that can increase or decrease the chances of developing cancer, as well as different cancer and their behavior. The book first starts on the genetic level, providing detail on cancer’s development from a genetic level and the highest risk factor. It then describes modern medicine used to combat cancer, to the factors that can either increase or decrease the probability of developing cancer. Finally, the book oversees the different techniques in medicine to combat cancer as well as provides detailed information about different cancer in existence. Being the tenth edition of the book, edited by three medical Doctors, released in 2016 by the organization Wolters Kluwer which has existed since 1836 makes this book a scholarly reference. This reference can describe cancer and provide broader information regarding the factors related to cancer used to develop machine learning algorithms.

Feng, Y., Yang, K., Sun, H., Liu, Y., Zhang, D., Zhao, Y., Shi, W., Lu, G., Zhang, Z., Jia, A., He, S., & Li, H. (2021). Value of preoperative gastroscopic carbon nanoparticles labeling in patients undergoing laparoscopic radical gastric cancer surgery. *Surgical Oncology*, *38*, 101628. https://doi.org/10.1016/j.suronc.2021.101628

This journal article provides insight on the effect that preoperative gastroscopic carbon nanoparticles labeling has on patients who are undergoing surgery for stomach-related cancer. The article then details how the data was gathered retrospectively. It was collected from a system containing electrical records and data from past surgeries done in their institution. It finally describes the results of it’s statistical analysis. This article was published by Elsevier, revised on 7 June 2021, and accepted on 11 June 2021 making this very recent journal article a scholarly reference. This article demonstrates the statistical aspect in determining the efficiency of certain methodologies which can be crucial to determine features within datasets that can potentially be utilized to develop machine learning algorithms.

Hadjiyski, N. (2020). Kidney Cancer Staging: Deep Learning Neural Network Based Approach. *2020 International Conference on E-Health and Bioengineering (EHB)*, pp. 1-4. https://doi.org/10.1109/ehb50910.2020.9280188

Article details the different stages of kidney cancer and how deep learning (a subset of AI) can determine the stage of kidney cancer to assist medical professionals in determining the correct treatment. Images extracted from CT scans of kidney cancer patients were taken and a deep learning algorithm was utilized to label, train, and test on the collected images. Then statistical analyses were used to determine the accuracy of the deep learning algorithm’s ability to classify the different stages of kidney cancer using images. This is an IEEE article published in the 2020 International Conference on e-Health and Bioengineering. These two factors (meaning that the article is backed by a large association related to all fields of science and that the article was published in a conference that may have required peer review) make this article a scholarly source. This is a clear example of the potential that AI has to aid cancer research.

Husaini, M. A. S. A., Habaebi, M. H., Hameed, S. A., Islam, Md. R., & Gunawan, T. S. (2020). A Systematic Review of Breast Cancer Detection Using Thermography and Neural Networks. *IEEE Access*, *8*, 208922–208937. https://doi.org/10.1109/access.2020.3038817

Using AI and deep learning to predict signs of breast cancer within images taken via thermography. Compares thermography techniques to other imaging modalities by providing its advantages and disadvantages. This article was published in March 2020 at an IEEE conference in New Delhi, India. This would make this article a scholarly reference as it is published by a multinational institution focused on research and innovation. This is a clear example where AI shows the potential to supplement new imaging techniques to identify cancer at its earlier stages.

IBM Cloud Education. (2020, June 3). *What is Artificial Intelligence (AI)?* Www.ibm.com. https://www.ibm.com/cloud/learn/what-is-artificial-intelligence

Contains basic information on Artificial Intelligence (AI). Can be utilized to define, and explain the origin of AI.

Iyer, V., Hima Vyshnavi, A. M., Iyer, S., & Namboori, P. K. K. (2019). An AI driven Genomic Profiling System and Secure Data Sharing using DLT for cancer patients. *2019 IEEE Bombay Section Signature Conference (IBSSC)*, pp. 1-5. https://doi.org/10.1109/ibssc47189.2019.8973020

Using genetics and sample images of patients with different melanoma (skin cancers) to train a model that would predict early signs of melanoma along with the gene associated with the melanoma type. This article also dabbles in information sharing to keep patient confidentiality secret. It was published by IEEE in the 2019 IEEE Bombay section Signature Conference. As IEEE is a multinational institution that focuses on research, this article can be of the scholarly variety. This is yet another example of how AI can be utilized to research and identify cancer.

Kulkarni, S., Bhat, S., & Moritz, C. A. (2019). Reconfigurable Probabilistic AI Architecture for Personalized Cancer Treatment. *2019 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1-7. https://doi.org/10.1109/icrc.2019.8914697

This article tackles the aspect of cancer as a disease caused by genetic defects by developing an AI model that can personalize treatment for patients under the circumstances. As IEEE is a multinational institution that focuses on research, this article can be of the scholarly variety. This is another example of how AI’s potential to personalize cancer patient’s treatments.

Moser, E. C., & Narayan, G. (2020). Improving breast cancer care coordination and symptom management by using AI driven predictive toolkits. *The Breast*, *50*, 25–29. https://doi.org/10.1016/j.breast.2019.12.006

This article tackles customized cancer treatment from primary health professionals through long periods of time by speaking about how AI can be used in cancer treatment to mitigate some issues that occur at different stages of care. This article was published by Elsevier on “The Breast” Volume 50, making this a scholarly article as it has been peer-reviewed. This article demonstrates the ideal situations where AI has great potential to aid Cancer research as well as its treatment.

Tew, K. D., & Fisher, P. B. (2019). *Advances in cancer research. Volume 142*. Academic Press. https://gcu-encore.iii.com/iii/encore/record/C\_\_Rb1373095\_\_SCancer%20\_\_P0%2C21\_\_Orightresult\_\_U\_\_X6?lang=eng&suite=def

This book shows advances in cancer research. It demonstrates the technology currently used to detect and treat cancer. This is a scholarly article because it was released on ScienceDirect, a platform that focuses on scientific research, and it was also edited by at least one medical Doctor. This provides further details on the advances of technology in cancer research and also illustrates how AI can boost the current advances in cancer research.

Yadav, K. K. (2018). How AI Is Optimizing the Detection and Management of Prostate Cancer. *IEEE Pulse*, *9*(5), 19–19. https://doi.org/10.1109/mpul.2018.2866354

This article provides a short description of prostate cancer for men, as well as some statistics associated with the mortality, costs, and diagnoses done yearly for prostate cancer in males. This is a scholarly article published in 2018 by IEEE (IEEE Pulse). this can be utilized to point out where AI has the most potential to aid in cancer research and its treatments.

Appendix B – Copyright Compliance

MIT License

Copyright (c) 2022 Wilson Peguero Rosario

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cui, Chunyan; Li Li; Cai, Hongmin; Fan, Zhihao; Zhang, Ling; Dan, Tingting; Li, Jiao; Wang, Jinghua. (2021) **The Chinese Mammography Database (CMMD): An online mammography database with biopsy confirmed types for machine diagnosis of breast**. The Cancer Imaging Archive. DOI: <https://doi.org/10.7937/tcia.eqde-4b16>

Milestone 2

Design Planning Summary

Cancer is an affliction that has existed since ancient times. Multiple solutions have been created to mitigate the harm that cancer does on the human body (such as surgery, chemotherapy, gene therapy, etc.). Although these solutions have assisted in mitigating cancer deaths, cancer remains as the second leading cause of death in the United States. Majority of the deaths are mainly caused by the late diagnosis in cancer (the point in which the tumors have spread across the entirety of the body). To assist in early diagnosis (while mitigate the yearly rate of misdiagnosis of 5%), machine learning algorithms can be employed. With machine learning, doctors can better observe and diagnose tumors as either benign or malignant.

**Project Deliverables:**

* Determine whether tumor is malignant or benign
* Provide probability or likelihood of tumor being malignant vs benign

**Overview of Model Pipeline Design**

To develop the pipeline, the data will be extracted from Google’s BigQuery open data sets from the NIH’s data commons website. After the query is done, a table containing links to the image data set will be downloaded by using python and the google-auth library to log in and access the images saved on Google servers. The images will then be loaded using the pydicom library as the dataset will contain a series of DICOM files (medical image format that contains textual information within the metadata of the image; DICOM stands for Digital Imaging and Communications in Medicine). The pydicom library will then be used to extract the textual data and the image, then TensorFlow’s Data API will be used to load and process the data. The image data will be processed using the preprocessing layer within the model while another preprocessing layer from TensorFlow will be used to convert textual data to numerical data. In the case that the textual data represents a label, the label will be encoded using TensorFlow’s OneHotEncoding functionality. This will allow the model to be trained without being misled. To visualize the data and observe patterns, one may grab a random sample to plot side by side with the different labels (i.e. a matrix of images at different stages of cancer). To analyze the model and interpret the results from the analysis, at least four models will be used. A base model with a single convolutional layer to observe the potential of Convolutional layers to make predictions, a model with preprocessing layers that will in turn normalize the data before processing to observe the effects of the preprocessing layer on the last model, a layer that interprets the contribution of the textual data to the explanatory variable and finally a composite model which takes in both processed image data and textual data to make predictions on the stage of cancer or whether the tumor is malignant or benign. A comparison between the metrics of all four models will be used and the predictions done will in turn highlight how the composite model may have its advantages over the independent models at making predictions or may suffer from the disadvantages of the worst performing model.

Detailed Model Pipeline Design

**Overview**

Cancer is the second leading cause of death within the United States. Although many cancers, such as cervical cancer, are completely treatable, many patients still die due to late diagnosis or even misdiagnosis. Although doctors have been able to identify and diagnose tumors based on measurements and medical images (i.e., Ultrasound, MRI, X-Rays, etc.), it can be difficult to determine with confidence whether a tumor is benign or malignant. The current rate of yearly diagnoses in cancer is 95% (1 in 20 people are misdiagnosed with cancer). Although it may be difficult for doctors to diagnose cancer on their own, machine learning algorithms can detect the most minuscule features related to the tumor in order to determine whether said tumor is malignant or benign. This may assist in decreasing the rate of misdiagnosis (which is currently 5% of patients diagnosed with cancer per year).

**The Data Sources**

The data sources will be the Cancer Imaging Archive (a set of public and private data sets containing hospital image data with annotations, diagnoses, etc.), The Imaging Data Commons (a more accessible version of the CIA which allows one to query and extract the data through Google’s BigQuery). These data sources are both backed by the NIH, meaning that the data has been curated and stems from reliable sources that use professional techniques to not only anonymize the data, but also to properly curate the data into a functional data set.

**Data types and Formatting**

The main data types are:

1. Image data
2. Textual Data

Both these data types will be loaded and extracted using pydicom and Google’s googleauth library from python to then process the images and the textual data for model training. The formatting of the data will be done through TensorFlow’s data api and machine learning layers. The images will undergo a processing layer called a convolutional layer for dimensionality reduction and feature extraction, while the textual data will be converted to numerical data.

**Data Cleaning**

Before extracting the features from the image, some images from the same sample may be removed and the data will be balanced as well. A separate data set containing image data with little to no description will be set apart from the data set used to train the data as one would expect to have data with similar textual annotations for each sample image provided.

**Initial Data Exploration and Visualization**

The initial data exploration will contain some metrics estimating the average word count of the annotations, the raw number of data points per label and a comparison of the images provided with the labels will be shown to demonstrate the difficulty of estimating the stage of cancer or whether a tumor is benign or malignant based on the image alone. The header metadata values selected to become part of the input for the model were selected based on the knowledge provided by studies in regard to factors leading to cancer. An example of this would be age (where for all genders the likelihood of having cancer increases as one ages), other factors include sex, weight, and more. Some other aspects related to the patient also impact likelihood that he or she may suffer from cancer, such as race, height, economic status, and education (these factors are more blanket in nature as they encompass diet, mean height, exposure to radiation through microwave, etc.). All these factors are evaluated based on the correlation matrix.

**Data Models and Nature**

There will be 4 models at the very least. The first three models will provide insight on the power of processing the image to make predictions, the impact of data augmentation and normalization, and the impact of feature extraction or dimensionality reduction. The first model will use the entire image to make a prediction, whereas the second model will use dimensionality reduction through convolutional layers to make a prediction, the third model will include only data augmentation and normalization and finally the last model will include a combination of the latter two models. The purpose of the last model is to not only save training time and model size, but also prevent the introduction of random noise through the image itself.

**Methodology**

The results will be interpreted based on the metrics provided by the TensorFlow library. Accuracy, loss in training, mean squared error, precision, recall, and other metrics will be used in tandem throughout the entire training procedure and testing procedure. A set of confusion matrices per model will be used to observe how the addition of certain factors may impact each model’s ability to recall and classify with precision. With the model complete, the user will upload the data through the Dash web app created. The image data and header data will then be extracted from the DICOM files uploaded to the app using two functions (one set to only extract the data from the DICOM file while another transforms the data and readies it for input). The application will then plot the data within six different charts portraying the data provided from the header files within the DICOM file together with the model’s prediction. At the bottom of the dashboard, is a table containing the metadata associated with uploaded DICOM file together with the prediction, and the probability of the tumor being benign or malignant.

**Configuration Changes**

To further develop the project some changes were in order. The first change to the pipeline is the manner in which data was fed to the system. As the machine learning model possesses two distinct inputs, data will need to be gathered within a dictionary before feeding into the machine learning model. As this is an online dashboard, not many great changes are required. The current API systems in relation to hospital EHR systems use the Fast Healthcare Interoperability Resources (FHIR) system that allows for the use of JSON outputs. This allows one to directly extract patient data and expand the categorical inputs of the machine learning model directly from the system. Meaning that any updates to the system or any historical validation of a patient may be used to further train the model and improve its accuracy. The change to this would be the manner in which one extracts the data, currently the data is downloaded as an extract from the database the related metadata is loaded from an excel or CSV file and finally transformed into a format that can be used for training and testing.

**Security**

As this application will be set locally to the environment, there will be no need for the application to meet any security standards. As a precaution the application will provide instructions on how to run the application locally similar other local plugins or applications (such as the OHIF viewer below):

[OHIF/Viewers: OHIF zero-footprint DICOM viewer and oncology specific Lesion Tracker, plus shared extension packages (github.com)](https://github.com/OHIF/Viewers)

The user or institution can activate a docker container image of the software and activate it into their local server for internal use only. Updates will be provided through GitHub or privately through the EHR system as a plugin.

**Hardware and Software Technologies**

The key hardware are the GPU, the memory, and the cpu, whereas the key software technologies are python, TensorFlow, pydicom, pandas, and numpy. Please observe the detailed explanation below:

1. The GPU (Asus TUF RTX 3090 TI)
   1. Although there are machine learning focused Graphics cards, the price range on the core performance is not worth buying for its functionality (the 3090TI provides great cooling solution while providing 24 GB for almost a third of the price of a machine learning focused RTX A6000 with 48GB of memory (The Costs for the NVIDIA RTX A6000 is $4,650.00 before taxes whereas the cost of the ASUS TUF 3090TI was $1613.31 after tax).
2. The CPU (AMD Ryzen 7 3800XT)
   1. The selection of the CPU is arbitrary as one used the GPU for the model training. The only component that would affect training in regard to the component affected would be the capability of the memory speed being locked due to the CPU being of a lower performance variety.
3. The Memory (Crucial Ballistix DDR4 4000MHZ)
   1. Required to transfer the model and data over to the GPU memory for training. The current size of the memory is 16 GB (two kits of 8GB) that may require expansion to 32GB to better match the size of the GPU memory.
   2. The speed of the memory was considered for allowing the user to comfortably transfer data without the memory speed being the bottleneck. Ideally, the best memory speed is somewhere between 3200-3600MHz with 4000MHz being able to sustain any heavy data processing tasks.

**Detailed Overview**

Graphical user interface, diagram, application

Description automatically generated

**Hardware and Software Technologies**

|  |
| --- |
| **HARDWARE AND SOFTWARE TECHNOLOGIES** |
| 1.NVIDIA RTX 3090TI GPU |
| 2.AMD RYZEN 7 3800XT |
| 3.16GB of RAM |
| 4. python 3.10 or Julia |
| 5.Tensorflow or Flux |
| 6. pydicom library |
| 7. Dash library |

|  |  |  |
| --- | --- | --- |
| **PROOF OF CONCEPTS** | | |
| **Description** | **Rationale** | **Results** |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |

Appendix A – Technical Issue and Risk Log

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ISSUES AND RISK LOG** | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| R | Model training | The training of the model may take majority of project time. | Use efficient algorithms and pre-storing training data for faster loading and training | Wilson Peguero Rosario | High | November 20th, 2022 | December 10th, 2022 | N/A |
|  |  |  |  |  |  |  |  |  |

Appendix B – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | [wpeguero/capstone: Capstone Project information. This contains the algorithm used for the Master's in Data Science Thesis. (github.com)](https://github.com/wpeguero/capstone) |
| **Hosting URL:** | *(if applicable).* |
| **Screencast**  **URL:** | *(if applicable).* |

Milestone 3

**System Entity**

The application will use the Dash library in conjunction with pydicom, pandas, and the TensorFlow libraries. The pydicom library will allow one to open and extract information from the uploaded DICOM images. The data collected using pydicom is transferred over to a pandas DataFrame object that in turn is used to run predictions through a TensorFlow Model object that was trained on previous data and designed based on the VGG-16 CNN model architecture. The Dash library is used to create an application class which builds the website. Ideally the website is hosted locally due to the size of the model (approximately 4Gb), but due to limitations made by the ISP provider, one is not able to provide public access to the website. A copy of the finalized code is provided as part of the system entity.

**Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Functional Requirement Number | Functional Requirement Description | Functional Requirement Function(s)/Class(es) |
| FR1 | The system must be able to load DICOM files. | capapp.parse\_contents()   * Function used to load the uploaded DICOM files for data extraction and transformation |
| FR2 | The system must be able to make predictions based on loaded DICOM files. | pipeline.extract\_data()   * Function which uses the pydicom library to read and extract useful data from the DICOM files. * Activated upon the user interaction with the predict button on the main page GUI.   pipeline.transform\_data()   * Function which uses basic python operations to prepare the data for model input.   models.tumor\_classifier()   * Function that uses TensorFlow’s Functional API to * Uses a combination of convolutional neural networks, pooling layers, and linear layers for final classification. * Outputs the prediction for use. * Activated once data is fully extracted from user selected files.   Pipeline.predict()   * Loads the model for making predictions |
| FR3 | The system must be able to display data extracted. | Capapp.update\_output()   * Displays the data within a dashboard format. * Contains a list of image files and some basic data related to the image file. * The list will also have a button to the right to allow the user to another frame that displays the targeted image. * Allows the download of CSV file. |

**Source Code Listing**

The third Iteration of the Capstone Project Application.

After failing to create the offline GUI through the use of Tkinter library and matplotlib, an online version of the same GUI has been replicated using the Dash library. This library uses the plotly library on the backend to develop plots and display them within html that the Dash library develops.

capapp.**parse\_contents**(*contents*, *filename*, *date*)

Load the content.

Function set to load the uploaded content and transform the data into a readable format for the machine learning model to make predictions.

**Parameters:**

* **contents** (*Unkown*) – Contains the uploaded file encoded into its 64-bit version. This is then decoded and read using the pydicom library for data processing.
* **filename** (*string*) – The name of the uploaded file. this does not impact the data processing aspect or the predictions, but it is necessary to point out which file the user is looking at.
* **date** (*datetime*) – Used to let the user know the date in which the file was uploaded. Will be used within report to download.

capapp.**update\_output**(*list\_of\_contents*, *list\_of\_names*, *list\_of\_dates*)

Load the main dashboard.

Models Module

This file will contain all of the actualized models created from the abstract model class(es) made within the base.py file.

models.**base\_image\_classifier**(*img\_height: float*, *img\_width: float*)

Basic Image Classifier for model comparison improvement.

…

A class containing a simple classifier for any sort of image. The models stemming from this class will function to only classify the image in one manner alone (malignant or non-malignant). This model will not contain any rescaling or data augmentation to show how significant the accuracy between a model with rescaling and data augmentation is against a model without any of these.

**Parameters:**

* **img\_height** (*float*) – The height, in pixels, of the input images. This can be the maximum height of all images within the dataset to fit a varied amount that is equal or less than the declared height.
* **img\_width** (*float*) – The width, in pixels, of the input images. This can also be the maximum width of all images within the dataset to fit a varied amount that is equal or smaller in width to the declared dimension.
* **batch\_size** (*int*) – One of the factors of the total sample size. This is done to better train the model without allowing the model to memorize the data.

**Returns:**

* **inputs** (*{img\_input, cat\_input}*) – Input layers set to receive both image and categorical data. The image input contains images in the form of a 2D numpy array. The categorical input is a 1D array containing patient information. This is mainly comprised of categorical data, but some nominal data.
* **x** (*Dense Layer*) – The last layer of the model developed. As the model is fed through as the input of the next layer, the last layer is required to create the model using TensorFlow’s Model class.

models.**base\_tumor\_classifier**(*img\_height: float*, *img\_width: float*)

Base Tumor Classification Algorithm.

…

A class containing a simple classifier for side-view image. The models stemming from this class will include rescaling for the sake and purpose of normalizing the data.

**Parameters:**

* **img\_height** (*float*) – The height, in pixels, of the input images. This can be the maximum height of all images within the dataset to fit a varied amount that is equal or less than the declared height.
* **img\_width** (*float*) – The width, in pixels, of the input images. This can also be the maximum width of all images within the dataset to fit a varied amount that is equal or smaller in width to the declared dimension.

**Returns:**

* **inputs** (*{img\_input, cat\_input}*) – Input layers set to receive both image and categorical data. The image input contains images in the form of a 2D numpy array. The categorical input is a 1D array containing patient information. This is mainly comprised of categorical data, but some nominal data.
* **output** (*Dense Layer*) – The last layer of the model developed. As the model is fed through as the input of the next layer, the last layer is required to create the model using TensorFlow’s Model class.

models.**tumor\_classifier**(*img\_height: float*, *img\_width: float*)

Complete Tumor Classification Algorithm.

…

A class containing a simple classifier for any sort of image. The models stemming from this class will include rescaling and data augmentation for the sake and purpose of normalizing the data.

**Parameters:**

* **img\_height** (*float*) – The height, in pixels, of the input images. This can be the maximum height of all images within the dataset to fit a varied amount that is equal or less than the declared height.
* **img\_width** (*float*) – The width, in pixels, of the input images. This can also be the maximum width of all images within the dataset to fit a varied amount that is equal or smaller in width to the declared dimension.
* **batch\_size** (*int \**) – One of the factors of the total sample size. This is done to better train the model without allowing the model to memorize the data.

**Returns:**

* **inputs** (*{img\_input, cat\_input}*) – Input layers set to receive both image and categorical data. The image input contains images in the form of a 2D numpy array. The categorical input is a 1D array containing patient information. This is mainly comprised of categorical data, but some nominal data.
* **output** (*Dense Layer*) – The last layer of the model developed. As the model is fed through as the input of the next layer, the last layer is required to create the model using TensorFlow’s Model class.
* *—*
* *\*Deprecated*

Pipeline Module

Algorithms used to process data before modeling.

…

A set of algorithms used to feed in and process data before used within the model. This will contain the data extraction from its rawest form and output the final form of the data set. The main source of data will be image related from the Cancer Imaging Archive.

pipeline.**calculate\_confusion\_matrix**(*fin\_predictions: DataFrame*)

Calculate the confusion matrix using pandas.

Calculates the confusion matrix using a csv file that contains both the predictions and actual labels. This function then creates a crosstab of the data to develop the confusion matrix.

Parameter(s)

fin\_predictions*Pandas DataFrame*

DataFrame containing the prediction and actual labels.

**returns:**

* **ct** (*Pandas DataFrame*) – Cross tab containing the confusion matrix of the predictions compared to the actual labels.
* **metrics** (*Dictionary*) – Contains the basic metrics obtained from the confusion matrix. The metrics are the following: - Accuracy - Precision - Recall - F1 Score

pipeline.**extract\_data**(*file*) → dict

Extract the data from the .dcm files.

…

Reads each independent file using the pydicom library and extracts key information, such as the age, sex, ethnicity, weight of the patient, and the imaging modality used.

**Parameters:**

**file** (*Unknown*) – Either the path to the file or the file itself. In the case that the .dcm file is already loaded, the algorithm will proceed to extract the data. Otherwise, the algorithm will load the .dcm file and extract the necessary data.

**Returns:**

**datapoint** – Dictionary comprised of the image data (numpy array), and the metadata associated with the DICOM file as its own separate *key:value* pair. This only pertains to the patient data and NOT the metadata describing how the image was taken.

**Return type:**

dictionary

**Raises:**

* **InvalidDicomError** – The file selected for reading is not a DICOM or does not end in .dcm. Set in place to stop the algorithm in the case that any other filetype is introduced. Causes an error to be printed and the program to exit.
* **AttributeError** – Occurs in the case that the DICOM file does not contain some of the metadata used for classifying the patient. In the case that the metadata does not exist, then the model continues on with the classification and some plots may be missing from the second page.

pipeline.**load\_data**(*filename: str*, *batch\_size: int*)

Load the data using tensorflow data set library.

…

Uses the os library and the TensorFlow Data api to load, batch, and process the data for training.

Parameter

filename*str*

Leads to a file containing the paths to all of the DICOM files as well as metadata.

batch\_size*int*

Factor of the length of the data set.

**returns:**

* **X** (*TensorFlow Dataset*) – Zipped dataset containing both image data and categorical data together.
* **y** (*TensorFlow Dataset*) – Data set containing the classifications of the data.

pipeline.**load\_testing\_data**(*filename: str*) → DataFrame

Load the data used for testing.

Loads a dataset to be fed into the model for making predictions. The output of the testing data will be comprised of a dictionary that can be fed directly into the model.

Parameter(s)

filename*str*

path to file containing the file paths to test data.

pipeline.**load\_training\_data**(*filename: str*, *first\_training: bool = True*, *validate: bool = False*)

Load the DICOM data as a dictionary. …

Creates a dictionary containing three different numpy arrays. The first array is comprised of multiple DICOM images, the second contains the categorical data as a vector, and the third contains the classification in numerical form.

**Parameters:**

* **filename** (*str*) – path to a file which contains the metadata, classification, and path to the DICOM file. Will also contain some sort of ID to better identify the samples.
* **batch\_size** (*int*) – Factor of the dataset size. Currently set to one as the standard for testing purposes.

**Returns:**

**data** – Dictionary containing the encoded values for the metadata and the transformed image for input to the model.

**Return type:**

dictionary

pipeline.**predict**(*data: DataFrame*, *model\_name: str*) → DataFrame

Make predictions based on dataset.

Extracts the image data and required categories for loading into the model.

pipeline.**rescale\_image**(*img: ndarray*) → ndarray

Rescale the image to a more manageable size.

Changes the size of the image based on the length and width of the image itself. This is to reduce the amount of computations required to make predictions based on the image.

Parameter(s)

img*Numpy Array*

array containing the raw values of images.

pipeline.**transform\_data**(*datapoint: dict*) → dict

Transform the data into an format that can be used for displaying and modeling.

…

Grabs the extracted data and begins transforming the data into a format that can be used for display in a dashboard as well as for modeling purposes.

**Parameters:**

**datapoint** (*dictionary*) – Contains the image and related metadata in *key:value* pair format.

**Returns:**

**datapoint** – same dictionary with the categorical data transformed into numerical (from text).

**Return type:**

dictionary

**Raises:**

* **AttributeError** – Indicator of the *key* does not exists.
* **KeyError** – Indicator of the *key* does not exists.

**Code Review**

Originally, the capapp was an application that focused mainly on a desktop application (that functioned as an executable). After review by the last professor and some advice from the fellow students of the last class, a decision was made to transition over to a web-based application. Another factor that influenced one towards the we-based application was also the failure of pyinstaller library to create the executable.

**Implementation Plan**

Overview

The purpose of this application is to save radiologists time while providing the best possible treatments to patients. Medical imaging is a technique through which radiologists are able to view the interior of the body and diagnose the patient based on what they have observed (MedlinePlus, 2019). The average radiologists observed (at least in 2017) approximately 20 to 100 scans a day, with each scan containing multiple images for review (Silverman, 2017). The use of DICOM files to make predictions will place the software application in a position where it must also ensure it meets the HIPAA compliance requirements regarding Personal Health Information (PHI) (Crotti, 2020). Once the software meets the HIPAA requirements for PHI, then hospitals will be able to use the software safely.

Installing the software will be done by downloading an executable file that can be found at a public Github page. As the application can be downloaded on any site, it provides administrators with the ability to distribute the application to all computers simultaneously. For future iterations, as the use of EHR systems increase, the software application will be an application developed within an EHR system such as EPIC (Evans, 2016).

Assumptions, Dependencies, Constraints

A lack of HIPAA compliance may be the biggest factor in delays to deployment. Before a product can be HIPAA compliant, multiple investigations must be conducted (Crotti, 2020). Running the risk of not securing the data may lead to not only investigations, but also possible loss of business and lawsuits (Crotti, 2020). Second to this, is the accuracy of the machine learning model. The daily rate of misdiagnosis caused by either systems or humans in radiology is between 3%-5% daily (Brady, 2016). To mitigate this, the model will have an accuracy upon its test of at least 95% with the goal (in terms of accuracy) being 98%. The third factor that may affect deployment is the popularity of EHR systems. As demand for standardization rises in hospitals, so does the need of EHR systems that not only neatly displays data, but also allows for ease download and transfer to other hospitals (Evans, 2016). Besides these three factors, the application should be ready for deployment.

Operational Readiness

Testing model predictions with regressive data in conjunction with comparison between models will be the main form of assessing deployment readiness. As spoken before, the model must be fully trained and saved before making the Graphical User Interface (GUI) into an executable file. This is because without the predictions, the entire software application itself becomes useless. One can also assess readiness based on the current market for offline medical software applications. Many locations within hospitals do not have wireless internet capabilities due to the noise and disturbance it may cause when in use with other medical devices, therefore offline GUIs are popular to use even today (Concordia University, 2015).

Data Conversion

The conversion of data will be based on the GUI itself. The user will first load the paths to the DICOM files that he or she wishes to diagnose, then the GUI application will begin extracting the DICOM data that it requires to make prediction and plot some charts. Once the main dashboard of the GUI is loaded, the user will be able to download a csv containing an array of the image used together with the patient metadata at the click of a button. Within the csv, will be a set of probabilities regarding whether the image should be classified as benign or malignant.

Phased Rollout

The first phase of the rollout will consist of a free version of the software without the ability to download the full report. Only the patient ID and predicted classification columns will be downloaded for the radiologist to update the diagnosis on his images. The second phase will include a purchasable and updated version that will allow the radiologist to download the complete report displayed within the application. Depending on the popularity of EHR systems, a port containing similar functionality to the offline application will be introduced as an extension for the EHR system. All of this will be run through either an on-premises server or alternatively through a cloud service.

Support

Support will be provided through online documentation within the desired website. The documentation will provide a guide on how to use the application while a dedicated email address will be provided for the sake of allowing interaction between hospitals and the company. In a future phase, a forum will be built that would allow users to comment on the issues that they may be encountering.

Release Planning

The first release will be considered as version 1.0. Any version before this release will be considered beta or experimental and only recommended for experimentation and feedback. After version 1.0, the product will be provided with both a free version and a paid version. In the case that the paid version does not reach targets, then a price range will be set in place while extending the ability for the user to extract more data from the DICOM file.

**Application Functionality and Execution**

The application can be accessed through the page [Web App - Unavailable (capstonetumorgui.azurewebsites.net)](https://capstonetumorgui.azurewebsites.net/)

Alternatively, the entire application is available within the following Github page:

[wpeguero/capstone: Capstone Project information. This contains the algorithm used for the Master's in Data Science Thesis. (github.com)](https://github.com/wpeguero/capstone)

As a third alternative, a zip folder has been sent as part of the project.

**User Guide**

This is the guide to using the application known as “The Capstone Project”. Once the application has been fully setup, the user may be able to access the website. The website will appear similar to the image below:

Graphical user interface, application

Description automatically generated

Click on the drag and drop or Select Files area to load multiple images (alternatively, the images can be dragged to the same area.

A computer screen capture

Description automatically generated with medium confidence

Select the image(s) and click open. Some time may pass before the images are displayed, but the following is part of the dashboard after the images load:

Chart, bubble chart

Description automatically generated

Below is the second half of the dashboard:

Shape

Description automatically generated

The export button can then be used to download a copy of the table which will display as Data.csv.

Appendix A – References

Brady, A. P. (2016). Error and discrepancy in radiology: inevitable or avoidable? *Insights into Imaging*, *8*(1), 171–182. https://doi.org/10.1007/s13244-016-0534-1

Concordia University. (2015, June 16). *Can personal devices interfere with hospital care? Study examines whether wireless transmitters represent a danger to patients*. ScienceDaily. https://www.sciencedaily.com/releases/2015/06/150616123921.htm#:~:text=New%20research%20helps%20define%20a

Crotti, N. (2020, November 9). *How does HIPAA affect medical devices?* Medical Design and Outsourcing. https://www.medicaldesignandoutsourcing.com/how-does-hipaa-affect-medical-devices/

Evans, R. S. (2016). Electronic Health Records: Then, Now, and in the Future. *Yearbook of Medical Informatics*, *25*(S 01), S48–S61. https://doi.org/10.15265/iys-2016-s006

MedlinePlus. (2019). *Diagnostic Imaging*. Medlineplus.gov; National Library of Medicine. https://medlineplus.gov/diagnosticimaging.html

Silverman, L. (2017, September 4). *Scanning The Future, Radiologists See Their Jobs At Risk*. NPR.org. https://www.npr.org/sections/alltechconsidered/2017/09/04/547882005/scanning-the-future-radiologists-see-their-jobs-at-risk

Milestone 4

**Components Testing**

|  |
| --- |
| ***Test Case Name:*** test\_pipeline |
| ***Priority:*** High |
| ***Module:*** pipeline.py |
| ***Test Objective:*** Test the functionality of the pipeline data extraction functions and assure that the output of the functions matches the input of the functions and algorithms on the models module. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step |  | Test Detail | Expected Results | Problem/Issue |
| 1 |  | The test\_extract\_data\_output function examines whether the output of the extract\_data function is a dictionary. | Datatype: dictionary | Dictionary datatype is required for model training and input. |
| 2 |  | The test\_transform\_data function examines whether the output of the transform\_data function has properly transformed the categorical data to its numerical representation. | The categorical values within the output are numerical in nature | Machine learning model observes categorical data as numbers (these are transformed into vectors of length n where n is the number of categories). |
| 3 |  | The test\_rescale\_image function examines whether the rescale\_image function rescales the image input by a factor of exactly two. Any larger reduction may cause a significant loss of data a lower factor may not be enough for data training. | Factor of 2 | The reduction in scale of an image must be exactly a factor of 2 |

|  |
| --- |
| ***Test Case Name:*** test\_model |
| ***Priority:*** High |
| ***Module:*** models.py |
| ***Test Objective:*** Examine the efficacy of the models as well as its ability to provide proper output. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step |  | Test Detail | Expected Results | Problem/Issue |
| 1 |  | The test\_model\_compilation function observes whether machine learning model can be loaded and set up for training. This function only examines whether the model has been successfully compiled or not | Model can make predictions post compilation | The model cannot compile for training. |
| 2 |  | The create\_test\_data function works to create a data set for testing different aspects of the model. | Test data set is created | Require data for testing the output aspect of the machine learning model. |
| 3 |  | The test\_model\_output function uses the create\_test\_data function to examine whether the tumor\_classifier function can create a model that outputs a list of probabilities based on classifications. | The output is a list of probabilities | The output of the model must be a list of probabilities to observe the model’s confidence. |
| 4 |  | The test\_model\_score\_probability function works to assure that the sum of the list of probabilities within the output equate to 1.00 | Sum of probabilities = 1 | The output is a list of probabilities that show the probability that the image contains a malignant or benign tumor. |
| 5 |  | The test\_model\_accuracy function iterates through a set of examples, creates a confusion matrix and calculates the accuracy using the pipeline module’s calculate\_confusion\_matrix function. It then asserts that the model accuracy is higher than or equal to 0.90. | Accuracy of model higher than or equal to 0.90 | The accuracy of the model must be at the very least 90% to compete against radiologists. |

**Requirements Testing**

There are a total of three general components, two of which will require testing. The three components are the data pipeline, the models module and the dashboard. Out of the three, only the data pipeline module and the models module require testing. Although the dashboard is provided and created, the main purpose of the algorithm is to develop a machine learning model that will classify image data, providing an api that allows one to use the machine learning model to make the predictions and display them on any dashboard app, such as python’s Dash library, or even Microsoft’s powerBI or Tableau will allow any institution interested to create their own machine learning model while easily displaying the results through the use of the data pipeline module.

|  |
| --- |
| Component:  Ability to Extract, Transform, and Load data for training |
| Name of Developer:  Wilson B. Peguero Rosario |
| Name of Reviewer:  TBD |
| Checklist   |  |  |  | | --- | --- | --- | | Type | Pass | Comments | | Functionality | **Passed** | Pipeline module is able to extract the data into a dictionary whose data is then transformed and loaded into a dataset object for model training. | | Versatility | **Passed** | The three functions to project ETL unto the data are interchangeable (meaning that they can be replaced independently of each other) while allowing one to save the training data for loading at another time. | | Adaptability | **Passed** | Able to select whether the model the data has been transformed for the first time or not to allow faster loading of data. | |

|  |
| --- |
| Component:  Ability to create machine learning model |
| Name of Developer:  Wilson B. Peguero Rosario |
| Name of Reviewer:  TBD |
| Checklist   |  |  |  | | --- | --- | --- | | Type | Pass | Comments | | Functionality | **Passed** | Model architecture can be used for any image or type of cancer. | | Performance | **Failed** | The machine learning model’s accuracy when viewing test data has increased from 63% to 71% by changing the from\_logits parameter to True. This still fails the 90% accuracy threshold. | |

**System Testing**

Text

Description automatically generated

**Figure 1:** Results of testing using the pytest framework.

Based on the testing set up for the model and the pipeline modules, the majority of the algorithm is ready for production (with the exception of the model accuracy). The ETL aspect of the data pipeline was tested together with the model compilation, training, input, and output. As mentioned before, the purpose of the dashboard is for standard display, but the purpose of the pipeline and models API is to create a machine learning model and feed the data to any dashboard application or library.