WGU C964

CAPSTONE PROJECT PROPOSAL

Derek Brown

3/1/2023

Part A: Project Proposal for Business Executives 3
Letter of Transmittal 3
Project Proposal 4
Overview 4
Problem Summary 4
Application Benefits 4
Application Description 5
Data Description 5
Objectives and Hypotheses 5
Project Methodology 6
Funding and Stakeholders 6
Data Precautions 7
Developer's Expertise 7
Part B: Project Proposal 8
Problem Statement 8
Customer Summary 8
Existing System Analysis 8
Data 9
Project Methodology 9
Project Outcomes 10
Implementation Plan 10
Evaluation Plan 10
Resources and Costs 11
Timeline and Milestones 12
Part D: Post-implementation Report 14
Organization Vision 14
Datasets 14
Data Product Code 15
Objective Verification 16
Effective Visualization and Reporting 16
Accuracy Analysis 18
Application Testing 18
Application Files 18
User Manual 20
Summation of Learning Experience 21
Sources 21

Part A: Project Proposal for Business Executives

Letter of Transmittal

Derek Brown, Senior Software Developer
Kosmos Logic LLC
4243 Starlight Lane,
Nova Stella, NS, L1G 4T0

March 1, 2023

Dr. Isaac Newton, CEO, and Founder
Heavy Matters Medical Group
1248 Center Street,
Gravity Falls City, ON, H3A 5Y0

Dear Dr. Newton,

RE: Software Solution for Alzheimer's detection

We are working on a proposal for a project that will be very beneficial to your company. The proposal addresses the task of manually reviewing MRI scans to diagnose Alzheimer's. This task has historically required extensive manual review and many hours of work. This proposal will explain how that process might be automated, which will result in more rapid results and lower costs overall.

The premise of the proposal is to use machine learning and artificial intelligence to automate the process of reviewing thousands of MRI scans to detect the presence of Alzheimer's more efficiently. This project has the potential to reduce the amount of time needed to review these scans by approximately 66% according to data provided by some of our previous clients after implementing similar projects. This project also has the potential to be expanded to many other areas to assist in the diagnosis and detection of the disease once this initial project is completed.

The estimated cost of this project will be an initial amount of $14,980 to create the proposed solution, and an estimated $2,000 per month thereafter for maintenance, support, and so on. The project is expected to save around $4,000 per month per user, which will offset the initial costs after around four months of usage.

Our company has been developing machine learning solutions for over 10 years for hundreds of clients around the world. We have a solid understanding of the challenges posed and how we can overcome them. I look forward to hearing back from you soon.

Regards,

Derek Brown

Derek Brown, Senior Software Developer
Kosmos Logic LLC
derekbrown@kosmoslogicllc.com

Project Proposal

Overview

The project will use machine learning and image recognition to assist in the detection and diagnosis of Alzheimer's using MRI scans. The project will scan the images and return a prediction of either "Positive" or "Negative" for each image, and all the results will be stored in a report.

Problem Summary

It has come to my attention that your company primarily relies on manual review for most diagnoses of MRI scans, including diagnoses for Alzheimer's. Manual review of MRI scans is a time-consuming and very inefficient process. The process is also subject to human error. This process should be automated to make the diagnoses more efficient and accurate. This project will use machine learning to create a solution that is capable of analyzing MRI images very quickly and with high accuracy. This will result in a faster turnaround, which will help your business to deliver a great experience to your patients, while also decreasing the costs associated with these diagnoses. This will improve your organization's ability to meet the needs of both the patients and the organization simultaneously. To help your organization accomplish these goals, our company proposes that we will create and deliver a software solution capable of making accurate predictions quickly. We will include all necessary documentation necessary for your medical professionals to easily use the solution within a matter of minutes. This solution has the potential to greatly expedite and enhance your organization's processes.

Application Benefits

The primary needs of your organization are to meet the needs of your patients while also reducing costs where possible. This solution will benefit your organization by achieving both of these goals at the same time. The solution we are proposing would reduce the time needed to complete the review of MRI scans by roughly 66% according to data voluntarily provided by our past clients, with some clients reporting a reduction of as much as 80% at times, which is remarkable and very beneficial for all parties involved. This makes the diagnosis considerably faster, which directly benefits the patients. It also decreases the costs of these processes, saving an estimated $4,000 each month for each user (this estimate is explained in the “Funding and Stakeholders” section below). In addition, this will decrease the workload for the medical professionals involved, which leads to lower stress levels and better overall performance by the professionals (which is corroborated by the reports from our past clients). Our clients have indicated an increase in overall productivity by roughly 15% once these solutions were set in place. That means in addition to saving time, these professionals are generally more productive following the implementation of solutions like the one we are proposing. In summary, the solution proposed has the potential to improve the patient experience, reduce costs, lower stress levels, and increase productivity for your organization.

Application Description

The project will utilize a machine learning algorithm to train a prediction model to recognize the patterns and symptoms of Alzheimer's which may be present in an MRI scan. This model will be trained using a form of supervised reinforcement. The data used to train the model will be predetermined as either a positive or negative diagnosis for Alzheimer's. This will allow the model to train effectively and identify the patterns associated with each. It will be able to analyze potentially hundreds of images in minutes, which would likely take hours of work for a manual reviewer. The report generated can then be used by the reviewer to get an overall picture of the situation, and they can focus their efforts on other important factors that will determine which diagnosis is ultimately given.

Data Description

The data will be obtained online using a repository of datasets called kaggle.com. The dataset which will be used includes 6,400 images of MRI scans which are labeled in four categories, depending on the severity of Alzheimer's. The data includes both an image and a descriptor for the image. The images in the dataset will be independent variables that act as the input for our model, and the predictions made by the model will be dependent variables (they are based on the input). The model is not expected to have any outliers, since the dataset is relatively homogeneous and nominal (I.e. it is qualitative, not quantitative). However, there may be some images that present issues due to the file being corrupted, the image is the wrong size, etc. Corrupted images must be removed, while images that are in the wrong format or size may be modified accordingly to the correct format or size. Images will be preprocessed before training to ensure they can be used effectively. The biggest limitation of this dataset is the size. A dataset of 6,400 images is relatively small when it comes to training a machine-learning model, and our dataset will be a subset of this. However, the subset should be adequate for the initial solution.

Objectives and Hypotheses

The main objective of this project is to make the diagnosis of Alzheimer's more efficient. This objective can be met by accomplishing three primary goals: ease of use, accuracy, and fast performance. To ensure ease of use, a simple interface will be used, and documentation will be included to assist users. To promote accuracy, the model is intended to meet or exceed specified accuracy thresholds initially and will be trained on larger datasets in the future which will increase accuracy. To ensure fast performance, the model will be designed using efficient algorithms to rapidly identify key elements of the images and quickly produce results. Using machine learning and a convolutional neural network (CNN), we will be able to create a model that can make accurate and quick predictions about whether or not an image depicts an MRI scan in which Alzheimer's is present. The model will be trained and evaluated to ensure accuracy and speed. The initial accuracy goal for the model will be above 80% and the initial speed goal will be under 2 seconds per image.

Project Methodology

The chosen methodology will be the waterfall model. This methodology was chosen because this project has a very clear path and certain steps that should be taken before other steps (for example, data collection must occur before the model can be trained). The waterfall model is ideal because of its sequential design. The waterfall model can be broken into five components: requirements, design, implementation, verification, and maintenance. Each of these is outlined in greater detail below:

Requirements: to address the needs of the organization to provide patients with timely and accurate diagnoses, the software has three main requirements - it should be easy to use, it should be fast, and it must make accurate predictions. To break these down further, ease of use will be encouraged by designing an interface and supporting documentation to allow users of any technical skill level to be able to use the software in minutes. The speed will be measured on a per-image basis and must meet a threshold of fewer than 2 seconds per image for acceptance. The accuracy of the model will also be measured and must meet or exceed a threshold of 80% accuracy for acceptance.

Design: the solution will be designed as a standalone program that can be set up on a Windows computer with Python installed. The solution will use a machine learning algorithm to train a model that is capable of predicting whether or not an MRI scan depicts Alzheimer's or not. The interface will consist of some buttons that provide functionality. This will include a button to retrain the model, a button to display the accuracy of the current model, a text field that allows the user to input a folder from their computer with a button allowing users to select a folder from their system, a button to make predictions for all the images in the folder which will produce a table displaying the results, and a button to show the percentage of "Positive" predictions and "Negative" predictions using a pie chart.

Implementation: this will consist of data collection and coding the software solution. The data will be collected and prepared for the solution, and the model will be trained. The user interface will be created and implemented to work with the model.

Verification: the software will be tested against the requirements specified above (2 seconds per image for speed, and 80% accuracy). If the requirements are not met, the software will be modified accordingly until it meets the requirements.

Maintenance: included with the software, documentation will be provided which will detail how the software should be maintained and how it can be improved. The software will include functionality to retrain the model, which will allow data to be added. This means that the organization will be able to add its data in the future, which will increase the accuracy of the model in future updates.

Funding and Stakeholders

The funding required for this project is estimated to be $14,980. This total includes the salaries of the software developer ($4,500), the data engineer ($2,750), and the project manager ($3,000) for this project and the cost of the computers and software licensing needed during initial development (totaling $4,730). The initial development cycle is expected to last 5 weeks, from the start of the project to its deployment. In addition, it is estimated to cost roughly $2,000 per month to cover maintenance, support, hardware, and personnel costs. The software used to develop the code for the project will not have an impact on the project's budget, as the PyCharm IDE is free. However, software licenses for Microsoft Suite will be required to generate the documentation, which will cost a total of $30 for the duration of the project for all three licenses (included in the figure above).

The expected impact this will have on the stakeholders is mostly positive. While some stakeholders may be hesitant to adopt technology in health-related areas, I hope that they will quickly see the positive impact it has, and this may serve as a springboard to implement similar solutions in other areas in the future. The expected benefit of this solution is the time saved by automating the review of MRI images, which is expected to save an estimated average of 2 hours per day per manual reviewer (doctors, specialists, and other medical professionals). At an average salary of $100 per hour for medical professionals, this would save about $6,000 per month. Including the $2,000 maintenance cost each month, this means a net of $4,000 would be gained if just one user uses the solution in a given month. This means that after about 4 months, the initial costs for the project would be covered, and the investment into this project would be realized. Those numbers are for a single user - if multiple users utilize this software solution, the savings would increase even more (although support costs are likely to increase as well, at a rate of less than $2,000 per month). In addition, the patients will receive a more timely diagnosis and the doctors and professionals will have a decreased workload and lower stress levels. All in all, there are many benefits to this solution for all stakeholders involved.

Data Precautions

Data security is a top priority. The data used to initially train the model will not include any identifying information for any individuals and thus requires no additional security measures. It is a public dataset. However, the software and any data created by the software will need to be handled with care, since this software will be working with medical data, which may include sensitive patient information. To ensure that this information is kept secure, the software itself will not transmit any information to any other computer systems or networks in any way. All data processed by the software should be housed solely on the computer on which the software is installed and run.

Developer’s Expertise

As a Senior Software Developer for Kosmos Logic LLC, I am very familiar with data engineering and software development practices. I have over 8 years of experience in software development and data engineering and a Bachelor's degree in Computer Science. I have worked on various projects involving machine learning and I have a solid understanding of the process and how to achieve the goals. In addition to my expertise, I will be working with a team of talented developers and engineers, some with more experience than myself. We will be consulting one another regularly to ensure that this solution meets the needs of the company.

Part B: Project Proposal

Problem Statement

Heavy Matters Medical Group currently uses outdated methods to analyze MRI scans and detect Alzheimer's. They use manual review, which is very slow and subject to human error. This process should be automated to make the diagnoses more efficient and accurate. This project will use machine learning to create a solution that is capable of analyzing MRI images very quickly and with high accuracy. This will result in a faster turnaround, which will help your business to deliver a great experience to your patients, while also decreasing the costs associated with these diagnoses. This will improve your organization's ability to meet the needs of both the patients and the organization simultaneously. To help your organization accomplish these goals, our company proposes that we will create and deliver a software solution capable of making accurate predictions quickly. We will include all necessary documentation necessary for your medical professionals to easily use the solution within a matter of minutes. This solution has the potential to greatly expedite and enhance your organization's processes.

Customer Summary

The client in this case is Heavy Matters Medical Group, which consists mostly of medical professionals. These professionals take care of hundreds of patients each day. In the medical field, the time it takes to make a proper diagnosis is crucial to ensuring quality care for patients. The solution we are proposing will be invaluable in speeding up the process of analyzing MRI scans for Alzheimer's and helping them to give the patients the care they deserve. By using machine learning, we will be able to provide accurate and timely results, which is estimated to reduce the time needed to review these images by 66% based on the data provided by previous clients.

Existing System Analysis

Currently, the MRI images are created and saved on a computer, which requires a medical professional to review the images to make a definitive diagnosis. The process is slow and cumbersome and does not utilize the power of their computer systems. The solution we are proposing will allow this process to be mostly automated, which will reduce the time needed by these professionals to reach a diagnosis. It will improve their capabilities and quality of care and will lower the workload for these professionals who are already working in a very stressful field. And on top of that the estimated savings from implementing this solution are around $4,000 per month. This estimate was reached using the following analysis: the expected benefit of this solution is the time saved by automating the review of MRI images, which is expected to be an average of 2 hours per day per manual reviewer (doctors, specialists, and other medical professionals). At an average salary of $100 per hour for medical professionals, this would save about $6,000 per month. Including the $2,000 maintenance cost each month, this means a net of $4,000 would be gained if just one such user uses the solution in a given month. This means that after about 4 months, the initial costs for the project would be covered, and the investment into this project would be realized. Those numbers are for a single user - if multiple users utilize this software solution, the savings would increase even more. In addition, the patients will receive a more timely diagnosis and the doctors and professionals will have a decreased workload and lower stress levels. This solution will address the current shortcomings of their processes while saving money in the long term.

Data

The data will be obtained online using a repository of datasets called kaggle.com. The raw dataset which will be used consists of 6,400 images of MRI scans which are sorted into four categories, depending on the severity of Alzheimer's. This data will be reorganized into two categories - "Positive" and "Negative." We will narrow down the dataset to 5,000 total images, with a 50% split for each category. The images in the dataset will be independent variables that act as the input for our model, and the predictions made by the model will be dependent variables (they are based on the input). The model is not expected to have any outliers, since the dataset is relatively homogeneous and nominal (I.e. it is qualitative, not quantitative). However, there may be some images that present issues due to the file being corrupted, the image size, issues with the format, etc. Corrupted images will be removed, while images that are in the wrong format or size will be modified accordingly to the correct format or size. The size we will use for images is 128x128 pixels for the training data. Additionally, the images need to be converted to the RGB color format. Once the images are processed and a label is assigned, they can be used to train the model.

Project Methodology

The chosen methodology will be the waterfall model. This methodology was chosen because this project has a very clear path and certain steps that should be taken before other steps (for example, data collection must occur before the model can be trained). The waterfall model is ideal because of its sequential design. The waterfall model can be broken into five components: requirements, design, implementation, verification, and maintenance. Each of these is outlined in greater detail below:

Requirements: to address the needs of the organization to provide patients with timely and accurate diagnoses, the software has three main requirements - it should be easy to use, it should be fast, and it must make accurate predictions. To break these down further, ease of use will be encouraged by designing an interface and documentation to allow users of any technical skill level to be able to use the software in minutes. The speed will be measured on a per-image basis and must meet a 2-second per-image threshold for acceptance. The accuracy of the model will also be measured and must meet or exceed a threshold of 80% accuracy for acceptance.

Design: the solution will be designed as a standalone program that can be set up on a Windows computer with Python 3.10 installed. The solution will use a machine learning algorithm to train a model that is capable of predicting whether or not an MRI scan depicts Alzheimer's or not. Specifically, the algorithm will use a convolutional neural network to achieve this. The interface will consist of some buttons that provide functionality. This will include a button to retrain the model, a button to display the accuracy of the current model, a text field that allows the user to input a folder from their computer with a button allowing users to select a folder from their system, a button to make predictions for all the images in the folder which will produce a table displaying the results, and a button to show the percentage of "Positive" predictions and "Negative" predictions using a pie chart.

Implementation: this will consist of data collection and coding the software solution. The data will be collected and prepared for the solution, and the model will be trained. The user interface will be created and implemented to work with the model.

Verification: the software will be tested against the requirements specified above (2 seconds per image for speed, and 80% accuracy). If the requirements are not met, the software will be modified accordingly until it meets the requirements.

Maintenance: included with the software, documentation will be provided which will detail how the software should be maintained and how it can be improved. The software will include functionality to retrain the model, which will allow data to be added. This means that the organization will be able to add its data in the future, which will increase the accuracy of the model in future updates.

Project Outcomes

The following will be delivered at the end of this project:

- The code to train the machine learning model.
- Code to create a user interface to interact with the model.
- The data used to train the model.
- Documentation for the data and software solution.
- A user manual.

Implementation Plan

Implementation of the solution will follow the steps below:

1) Data collection - obtain the data from the predetermined source (kaggle.com).
2) Complete programming for both the model and the user interface.
3) Complete data preparation and cleaning to prepare for training and testing the model.
4) Train and evaluate the model, making adjustments as needed.
5) Finalize and review the documentation and solution.
6) Deliver the deliverables to the client.

These steps will be outlined in greater detail in the “Timeline and Milestones” section below.

Evaluation Plan

The method to validate that the solution meets the requirements and needs of the customer is to utilize the model and use supervised learning. It will require the tester to use some test samples with predetermined labels to verify that the model meets the 80% threshold for accuracy. Another metric we will use to evaluate the model is speed, which must be no greater than 2 seconds per image. These will be verified using multiple systems before deploying the solution. The solution will then be validated by deploying it to a pilot set of computer systems before being implemented across the company. If the model does not meet the needs of the users or fails to meet the requirements, it will need to be modified either directly (modifying the training code) or indirectly (modifying the data used to train the model) and retrained. The interface includes features to retrain the model and since the code and training data will be available, it can readily be tweaked as needed.

Resources and Costs

The solution will be programmed using Python 3.10 using the PyCharm IDE on Windows 10. The only cost associated with the development environment is the Windows license, which will be included in the price of the hardware (it comes preinstalled), and the Microsoft Suite license (for documentation). These costs are listed below. The human resources needed include a project manager, who will act as the communication between the development team and the customer, a software developer, who will produce the code to train the model and will create the user interface, and a data engineer who will collect the data and prepare the data for the model. All three will also be involved in creating the documentation. See the table below for a breakdown of costs for initial development. In addition to the costs listed below, it is estimated to cost approximately $2,000 per month to cover regular maintenance, updates, and technical support once the solution is implemented.

Resource	Description	Cost
Software Developer 1	Implementing the code and software design for the project. Testing the model and review. Code documentation.	$45/hr X 100hrs = $4,500
Data Engineer 1	Collecting, cleaning, and preparing training and test data. Model review. Data documentation.	$50/hr X 55 hrs = $2,750
Project Manager	Coordinating the project, assigning tasks, reviewing, and delivering the final deliverables.	$60/hr X 50 hrs = $3,000
Laptop for Software Developer 1	A laptop capable of coding, training, and testing the model.	$2,000
Laptop for Data Engineer 1	A laptop capable of data collection and cleaning	$1,500
Laptop for Project Manager	A laptop capable of running project management software	$1,200
Microsoft Suite License	To create/edit documentation	$10/Mo X 3 Licenses = $30
	Total	$14,980

Timeline and Milestones

The following timeline offers a brief overview. More details for each are contained in the "Milestone Schedule" below.

March 13 - Project approval

March 15 - Project kickoff with a meeting

March 16 - Work begins

March 22 - Model training

March 29 - Model testing and second team meeting for a status update

April 3 - Solution review

April 10 - Final review (of the entire project)

April 12 - Third team meeting for a status update

April 17 - Solution and deliverables are delivered

Milestone Schedule

Milestone/ Event	Start	End	Tasks	Prerequisites/ Dependencies	Assigned Team Members	Total Hours
Project Approved	March 13	March 14	1. Project Approved	None	Project Manager (PM)	0
Project Kickoff	March 15	March 15	2. First team meeting	Task 1	All team members	5
Work Begins	March 16	March 22	3. Data collection and preparation 4. Code for model training 5. Code for UI	Task 2	Task 3 - Data Eng. Tasks 4 and 5 - Soft. Dev	60
Model Training	March 22	March 29	6. Training model 7. Data documentation 8. Second team meeting	Tasks 3, 4, and 5	Task 6 - Soft. Dev Task 7 - Data Eng. Task 8 - All	45
Model Testing	March 29	March 31	9. Model is evaluated 10. Functional	Task 6	Task 9 - Data Eng. Task 10 - Soft. Dev	30

			testing			
Solution Review	April 3	April 7	11. Code documentation 12. Solution evaluation	Tasks 9 and 10	Task 11 - Soft. Dev Task 12 - PM	30
Final review	April 10	April 14	13. Documentation review (all) 14. Final team meeting 15. Review against requirements	Tasks 7 and 11	Tasks 13, 14 - All Task 15 - PM	35
Solution Delivered	April 17	April 17	16. Solution is delivered	Tasks 12, 13, and 15	PM	0

Part D: Post-implementation Report

Organization Vision

The organization was struggling with the issue of modernizing its processes. In this case, the process to make a diagnosis of Alzheimer's was done manually, which required a medical professional to spend a long time looking over images to detect signs or symptoms of the condition. Our software solution was successful at solving the issue by utilizing machine learning algorithms to accurately and quickly make a prediction based on the images used as input. To use the model to make a prediction, first, the user should ensure that all relevant images are in a folder on a Windows machine, then run the Python file (app.py) which will create an interface to interact with the model. Note that Python 3.10 must be installed to run the file (instructions on installation of Python 3.10 are included in the user manual). Then either copy and paste the folder's address into the text field labeled "Select a Folder" or click on "Browse..." to select a folder using Windows Explorer. As long as the folder's file path is correct, then the user will be able to analyze the images and generate a report using the "Predict Results" button (an error message will appear at the bottom indicating if the folder could not be found if it is incorrect). The report will be displayed in a table, but will also be available as a CSV file which can be viewed in Microsoft Excel or a similar program capable of opening spreadsheets. The solution also includes other features which allow the user to retrain the model, load new data, and view statistics like accuracy and an overall breakdown of the results. Figure 1 below shows the user interface.

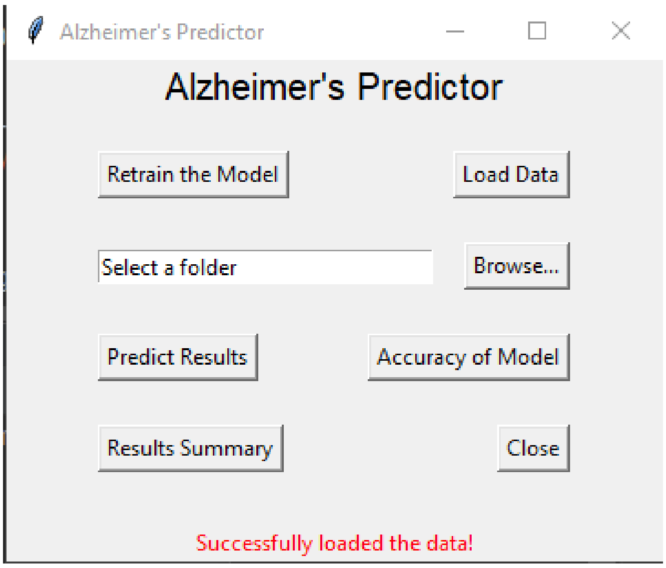


Figure 1. The user interface.

Datasets

The raw data consisted of images sorted into four folders based on the severity of Alzheimer's, including none, very mild, mild, and moderate. See Figure 2 below for an example. This dataset was obtained from kaggle.com. These were then manually sorted into folders "Negative" which consisted of the images in the first folder, and "Positive" which consisted of the other three folders. The modified dataset we used consists of a total of 500 images (split into training and testing data at a 4:1 ratio). The raw data must first be loaded using the "Load Data" button before it can be used to train the model. The images were resized to 128x128 pixels and reformatted from a BGR color format to an RGB format. Additionally, they were assigned a label ("Positive" and "Negative" based on the containing folder). The images and labels were then converted into an array. Finally, the data was saved as "loaded_data.pkl" which can be used by the model for training. The raw dataset is publicly available and can be found at the following link: https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images.

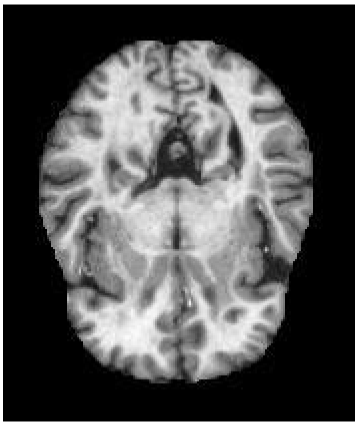


Figure 2. Example of the raw data

Data Product Code

The software can load data, train a machine-learning model, and make a prediction based on that model. In addition, the software can generate some statistics, including model accuracy, an overall breakdown of the results, and a detailed report showing the result for each image. To accomplish this, we used a convolutional neural network using several Python libraries including tensorflow, keras, numpy, and opencv to process the images and train the model. A convolutional neural network was chosen because of its ability to handle images well without losing information. The raw images are first loaded and processed to ensure they are compatible with the model, then the model processes the images by putting them through some filters in the neural network, and the model will attempt to identify the key elements of each image and will associate those elements with the label assigned to the image. This is used to make predictions about new, unlabeled images. The results of these predictions were saved to a document titled "report.csv" which is a spreadsheet that contains each image's name and its predicted diagnosis. Additionally, the results will appear in a table created using the matplotlib.pyplot library in Python.

Additional Python libraries such as tkinter, pandas, pickle, and csv were used to implement the user interface and other features (such as reading and writing to files). All of these were used because they provide the functionality needed in the given situation. For example, tkinter allowed us to create a simple user interface, the pandas library was used to assist in some data processing unrelated to the machine learning model, and both pickle and csv were used to open, save or modify files of different types.

Implementation of the solution followed the steps below:

1. The dataset was obtained from kaggle.com.

2. Programming for both the model and the user interface was completed.
3. Data preparation and cleaning to prepare for training and testing of the model were completed.
4. The model was trained and evaluated, adjustments were made as needed.
5. The documentation and solution were finalized and reviewed.
6. The deliverables were delivered.

To develop the machine learning portion of the solution capable of making fast and accurate predictions, we first needed to do two things: collect and prepare the data and write the code necessary to train the model. This was accomplished successfully and then training began. The model was evaluated for its accuracy upon completion of training and was retrained numerous times after making some adjustments. The model was evaluated by having it predict a set of test images, and the results were reviewed manually to confirm accuracy, and adjustments were made if the results were not satisfactory. Once the adjustments were made and we were satisfied with its capabilities, we moved on to the documentation.

Objective Verification

The objective of the project was to make the diagnosis of Alzheimer's more efficient by accomplishing three primary goals: ease of use, accuracy, and fast performance. Ease of use was addressed by creating a simple and intuitive user interface (see Figure 1 above). The accuracy was at almost 90%% based on training data. However, the accuracy will need to be monitored as the solution is rolled out to ensure that overfitting has not occurred. The accuracy of the model may not be reflective of its actual accuracy when predicting a diagnosis for new images. This can be measured by comparing the diagnosis given by the medical professional to the prediction from the model. This information will need to be provided by the professional. This will give a better idea of how accurate the model is in a real-world setting. The performance was well under the threshold we set based on our tests (usually measured in microseconds on our end), and we expect it to remain under 2 seconds per image, even on outdated or slow hardware.

Effective Visualization and Reporting

Visualization and reporting were crucial parts of development and provided a lot of feedback on how the model was performing and whether or not adjustments needed to be made. First, we have the accuracy graph, which shows how the model is improving with each epoch. This is an essential statistic to track because both learning too quickly and learning too slowly can indicate problems with the model. In some early models, for example, the accuracy started at 100% on the first and subsequent epochs due to mislabeling. The accuracy is depicted using a line graph, as shown in Figure 3 below.

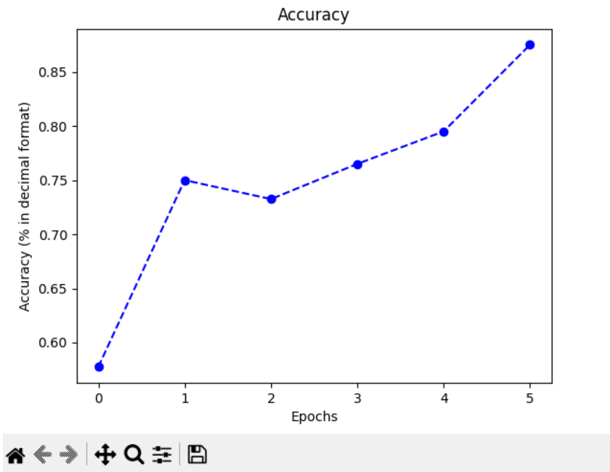


Figure 3. Graph of model accuracy

Another valuable visual report was the table of each of the images and the labels predicted by the model. This table was shown immediately following a prediction and is also available in the report.csv file (see Figure 4). This table helped when evaluating the model to ensure that it was able to make correct predictions. In some cases, the predictions were all the same, indicating that the model needed to be retrained. Having such immediate feedback was extremely valuable and helped in making the necessary changes promptly. This will also be valuable on the user end, as it will allow the user to look up each image flagged as a "Positive" by identifying the file names.

mild_166.jpg	Positive
mild_167.jpg	Positive
moderate_46.jpg	Positive
non_103.jpg	Positive
non_15.jpg	Negative
non_6.jpg	Negative
non_7.jpg	Negative
non_84.jpg	Positive
verymild_129.jpg	Positive
verymild_131.jpg	Positive

Figure 4. A table displaying the results

A third visual used was the pie chart used to display the distribution of each of the predictions. This is based on the predictions from the report.csv file and counts the number of each and displays a pie chart. See Figure 5 below. This helped us to see at a glance what was going on. This is more useful on the user end than on the developer end since it will be able to give a quick snapshot of what the images are predicted to be. Since MRI scans generally consist of hundreds or thousands of images, this will be extremely valuable as it will give a great look at the data as a whole.

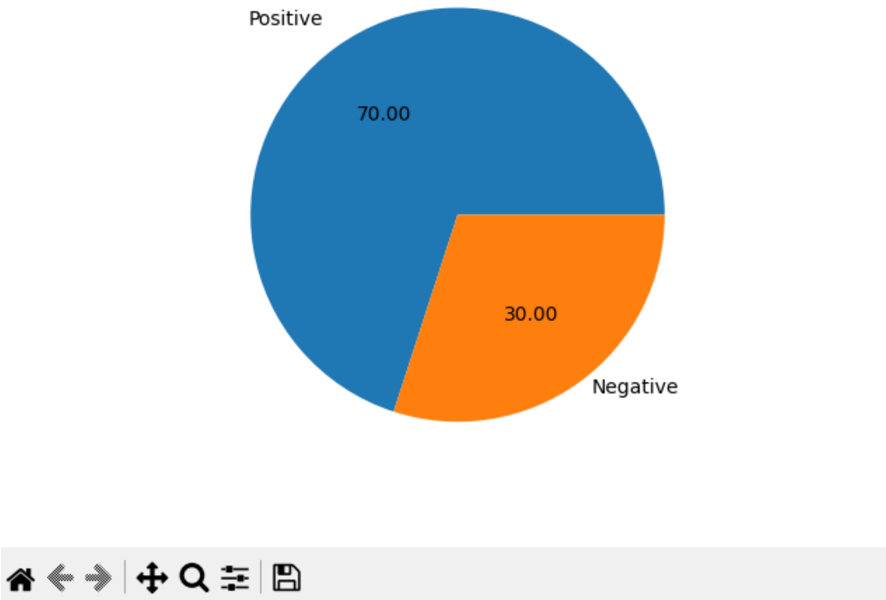


Figure 5. A pie chart showing the distribution of results

Accuracy Analysis

The metric used to analyze the accuracy of the model is the number of correct predictions divided by the total number of predictions, given as a percentage. The model's accuracy was at just under 90% with the training and test data, but it remains to be seen how accurate it will be when it is implemented in the environment in which it is intended to be used. The extent to which the current model will be able to generalize may vary somewhat. Because of this, the accuracy will need to be reassessed based on feedback given by the professionals using the software. In the future, a system may be implemented to allow the professional to provide such feedback within the software solution, which would help to improve the accuracy of the model.

Application Testing

The application was tested by running the program and manually testing each feature. When an error was encountered, the solution was inspected, and the cause of the error was identified and fixed. The interface also underwent a few renditions to make it intuitive, simple, and functional. The functions were tested as they were being developed to ensure that the result performed as expected. This occurred on an ongoing basis, but testing was also performed at the end to ensure that the intended functionality was there. Some of the modifications included adding confirmation messages to certain actions since it is possible to overwrite multiple different files using the interface.

Application Files

The file structure of the solution is outlined below (the structure of the bullet points indicates the heirarchy):

- **img-rec** - this is the root folder for the project. This folder contains the Python files and the resources folder.
    - **app.py** - this file contains the code for the main functionality and user interface
    - **train.py** - this file contains the code to load the data and train the model
    - **resources** - this folder contains various resources used by the code. Contains “data” and “testing” folders and the files “loaded_data.pkl”, “model_accuracy.pkl”, “report.csv”, and “trained_model.pkl”
        - **data** - this folder contains the data used to train and test the model
            - **test** - this folder contains the data used to test the model
                - **positive** - this contains 50 images identified as positive for Alzheimer’s
                - **negative -** this contains 50 images identified as negative for Alzheimer’s
            - **train** - this folder contains the data used to train the model
                - **positive** - this contains 200 images identified as positive for Alzheimer’s
                - **negative -** this contains 200 images identified as negative for Alzheimer’s
        - **testing -** this folder contains some test images used to evaluate the model's prediction capability
        - **loaded_data.pkl -** this contains all the processed data as a single file (the raw data files are found in the “data” folder)
        - **model_accuracy.pkl** - this is the accuracy data of the model, used to generate a line graph
        - **report.csv** - this contains the results of the prediction by the model

In addition to these files, the user will need to have Python 3.10 installed, including the following additional Python libraries:

- pandas
- tensorflow
- keras
- matplotlib
- numpy
- sklearn
- opencv-python

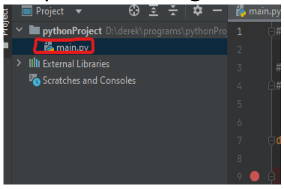
User Manual

Before running this software, you will need to have Python 3.10 installed on your system. Python 3.10 can be installed by going to this link and downloading either “Windows installer (64-bit)” or “Windows installer (32-bit).” This guide will also use the PyCharm IDE. Another IDE should also work, but instructions will not be provided for an alternative IDE. To install PyCharm, go here.

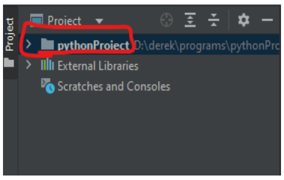
In addition to Python 3.10, the following libraries will need to be installed on the system if not already installed: pandas, tensorflow, keras, matplotlib, numpy, sklearn, and opencv-python. Users can install the required libraries in PyCharm using the “Python Packages” option near the bottom. Follow this guide for more details.

Once Python 3.10, PyCharm, and the libraries listed are installed, proceed to the steps below.

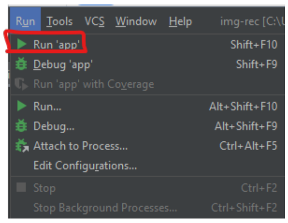
1. Download and unzip the folder. You can move the folder to your desktop for ease of access.
2. Run PyCharm and create a new project. Use all the default settings. You should now see a main.py file.
3. Try running the new project to ensure that it was created properly. If the file doesn't print a message like "Hi, PyCharm" to the console, there is an issue with your setup. Follow this guide if this is the case.



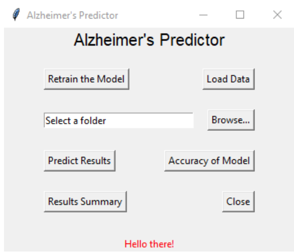
4. If everything is working correctly, delete the main.py file (you should see the file listed on the left-hand side - right-click and select “Delete”).
5. Right-click on the project folder and select “Open In” > “Explorer”



6. Drag and drop the folder “img-rec” from its current location into this project folder. You should now see the folder and its files where “main.py” formerly was.
7. Select “Run” and “Run ‘app’” at the top. See the image to the right.
   - If “Run ‘app’” is not an available option, try right-clicking on the filename for “app.py” in PyCharm and finding the “Run ‘app’” option.
   - If you have not properly installed the libraries listed above, you will get an error message when running the application. Follow this guide to install the libraries.



8. You should now see the user interface, as pictured to the right.



9. To make a prediction on a batch of images, select the “Browse...” button and select the folder on the computer where the images are stored.
   - Alternatively, you can type the folder into the text field, replacing the text “Select a folder”.
10. Click the “Predict Results” button. You will see a confirmation notice. Select “OK”
11. After the images have been processed, you should see a table displaying some results.
   - The table may not include all results for larger files; however, all the results will be found in the reports.csv file found in the resources folder (img-rec/resources/reports.csv). This file can be opened and viewed using Microsoft Excel and similar programs.

Additional features:

- Selecting the "Load Data" button will process all images found in the "data" folder and prepare them for retraining the current model. This should only be done if the data has been updated.
- Selecting the "Retrain the Model" button will allow the user to retrain the model. This should only be done if the current model needs to be updated with new data or is malfunctioning in some way, as this option will replace the current model with the newly trained model.
- Selecting the "Accuracy of Model" button will display a line graph showing the level of accuracy (based on the training data).
- Selecting the "Results Summary" button will display a pie chart that shows the distribution of the images predicted.
- The "Close" button closes the program.
- Pay attention to the status message at the bottom. If something goes wrong, it will indicate as much.

Summation of Learning Experience

Before this project, I had no experience working with machine learning, and it seemed like it would be a much more complicated task than it turned out to be. I don't have any professional experience with programming, only academic experience. My previous academic experience prepared me for this project by helping me to know how to break down the problem into smaller parts and work on one part at a time. It has helped me to understand the process of designing, implementing, and testing the software. To accomplish this task, I had to learn a lot about machine learning and how to implement it. I spent a lot of time looking over guides and explanations of image classification and related topics. I watched some videos walking through the process as well, which were helpful. This experience, while not the most intense project I've completed, was one of the projects that required me to learn the most. I learned a lot about convolutional neural networks and data processing. This experience has strengthened my commitment to lifelong learning by giving me a boost of confidence in my ability to learn and overcome challenges. I feel much more confident that I will be able to accomplish my goals as I continue to learn and gain new experiences.

Sources

Dubey, S., Alzheimer's Dataset (4 class of Images), https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images

Menon K., How To Build Powerful Keras Image Classification Models: Simplilearn (2022), https://www.simplilearn.com/tutorials/deep-learning-tutorial/guide-to-building-powerful-keras-image-classification-models