FIT3164 - Data Science Software Project

Final Project Report

Workshop: Thursday, 2pm - 4pm

Group: FIT3163_CL_04

Members: Elaine Liong (ID: 29942357),

Jack Ooi (ID: 29037077),

Vionnie Tan (ID: 30092809)

Word Count: 7883 words

1.2. Table of Contents

1.3. Acknowledgement

The completion and success of this project could not have been possible without the participation of many individuals.

Our group is extremely grateful for our project supervisors, Afsaneh and Uday, for their constructive feedback and generous acts of service in helping our team succeed with the project.

We would also like to extend and express our gratitude and appreciation for all of the FIT3163/FIT3164 teaching team for giving us continuous support, guidance, and providing us with the necessary resources throughout this project.

Lastly, the completion of this project wouldn’t be possible without the three members of Group FIT3164_CL_04 - Elaine, Jack and Vionnie. We would like to use this opportunity to thank each other for all the help and support we’ve given each other during the project timeline, especially during these difficult times! It was a pleasure to have been given the opportunity to work with each other on this project.

2. Introduction

For many years, cancer prevalence has been an ongoing issue among individuals in Australia. In 2020, Australia recorded over 150,000 new cases and 50,000 deaths caused by cancer (Cancer Council Australia, 2021). Deng et al. in 2015 reported that the five-year scale of cancer survivability rates between patients has been unsatisfactory due to side effects of relapses and metastasis from receiving treatment. As a result, researchers have been trying to improve the survivability rate of cancer-infected patients through applications of machine and deep learning. Although Australia has one of the highest cancer survivability rates in the world - with a mortality-to-incidence ratio (MIR) of 0.3, and in some cancers, survivability rates could reach as high as 90%, gastrointestinal cancer has one of the least developed methods of medication due to its difficulty to predict whether patients with Microsatellite Instability would respond well to immunotherapies based from their histology (Cheng et al., 2018; Cancer Council Australia, 2021; Kather et al., 2019). Through earlier detection of cancer, patients would be able to receive treatment earlier, which would, in turn, reduce the chances of malignant cells spreading uncontrollably in patients' bodies and increase their chances of survivability (Deng et al., 2015).

The above problem statement gives rise to the purpose of this report where we extensively discuss the aims of our project including the stakeholders involved, the main features of our software and its functionalities. Included also in this report, is the development of the software across two different semesters with its outcomes - that includes both implementation and shortcomings of the software produced and the different design methodologies such as tools, software, hardware and the algorithms implemented during the course of the project. Clear documentation of our software deliverables and quality assurance would be provided in this report. Lastly, this report would also cover a critical discussion of our software development that involves execution, deviation and limitations encountered.

3. Background

3.1. Project Aim

As mentioned in the problem statement above, our project aims to deliver a successful predictive model that allows for earlier detection of gastrointestinal cancers through image processing methodologies utilizing transfer learning techniques to classify Microsatellite Stable versus Microsatellite Instable or highly mutated cells provided in each image on pre-trained Convolutional Neural Networks, whose final product will be distributed publicly as a website that would be distributed publicly for health institutions to use. Our project could then be divided into two major parts - the predictive model component and the website. For the predictive model, our team aims to deliver a trained Convolutional Neural Network (CNN) model with high accuracy - preferably > 95%, as previously mentioned in our project proposal. The website component, on the other hand, would have all the basic functionalities required for a user to obtain a prediction result. This includes, but is not limited to, an upload image feature and a view prediction result button. In addition to this, our team aims to implement login and sign-up features so our users would be able to save and view their prediction results.

Due to uncertainties surrounding the project environment and in attempts to reduce the chances of our project failure, our team has also developed a minimum viable product (MVP) for this project to set as the minimum benchmarks for our deliverables at the end of the project timeline. This MVP includes a trained CNN model with a minimum of 60% accuracy for the predictive model and a simple website that would allow the user to upload and view their prediction result. Database functionality associated with user sign-up and login is not included in our minimum viable product. In addition to the MVP, documentation with regards to Project Management and Product related deliverables such as Business Case, Work Breakdown Structure, Source Code would also be provided by our team at project close-out.

3.2. Stakeholders

The small nature of our team environment emphasizes the importance of the impact a stakeholder has on the project's successes. Our team prioritizes the satisfaction of the stakeholders when delivering our product, as, during the close-out of the project, our team had planned to distribute our website to health institutions. This means that the stakeholders' feedback towards the project is extremely valuable to ensure that all requirements and acceptance criteria of the project are delivered by our website. As our team is using Agile Project Management Methodology, we have worked in a short burst of cycles - one week. As we have weekly workshop meetings with our stakeholders, it becomes convenient to receive their constructive feedback that would help in pointing us towards a better direction in the project.

3.3. Software Main Functionalities

For this project, our main deliverable was the website that contained the trained predictive model that users would access and predict their risk of contracting gastrointestinal cancer. This means that, for our end-users, the main functionalities served by our website is to allow users to upload, submit and view their prediction result of contracting gastrointestinal cancer. Our end-users are also able to easily navigate through our website and easily predict their risk through navigation buttons. Additional functionality of our software is that users can sign up to create an account and log in. This allows our users to access their latest results through their account creation and save their prediction results for future references and developments made to our software.

3.4. Software Main Features

As our software deals with the medical department, it is of utmost importance to have a clear depiction of features in our deliverables to ensure our end-users acceptance criteria and pain points are relieved. One of the notable features is that our website is easily accessible, with an easy-to-remember domain name and the flexibility to change domain names in future developments. The website is user-friendly, with a clear navigation bar that represents all the information relevant to our website as well as clear action to calls such as filling up the form and choosing a medical image to upload and view results. Our website is also accessible by everyone which ensures that our end-users wouldn't have any trouble accessing it. Through our rigorous testing, our website also has a relatively fast predicting time and works with any image of the accepted format.

3.5. Literature Review

Issues of Medical Data (Ooi et al., 2021)

There are many sources to obtain medical data such as images, interviews with the patients, physicians' notes and interpretation etc. (Delen, 2009). Delen (2009) illustrates the ideas of a variety of data and ethical and social issues of health data. According to Delen (2009), medical data is hard to obtain as the data are large, complex, heterogeneous, hierarchical, time series and of varying quality. Kumar et al. (2014) support the heterogeneity of data mentioned by saying raw health data is huge and heterogeneous. From the heterogeneity data point of view, Kumar et al.

(2014) state that medical data is written in different grammatical structures to explain the relations among medical entities. While for the ethical and social issues, Delen (2009) says that medical data is related to humans so that there is a chance where the patient's confidentiality might be breached and the possibility of ensuing legal action. Overall, many studies indicate that there are some issues when the process of medical data mining is taking place.

Transfer Learning

The idea of transfer learning goes hand-in-hand with data augmentation as it is believed to produce better accuracies in deep learning models (Bodavarapu et al., 2021). According to Gao & Mosalam (2018), machine learning heavily relies on large amounts of mostly labelled data and “useful data sometimes are very expensive and the worst case is that only few data can be collected.” Transfer learning became a very effective tool to diminish machine learning’s dependency on available data size (Gao & Mosalam, 2018). Transfer learning can be applied to deep Convolutional Neural Network (CNN) with pre-trained models’ generalization ability. Furthermore, Gao & Mosalam (2018) mentioned that transfer learning can tune part of the parameters from a pretrained model, which might lead to a good performance for the target data set. In transfer learning, there is a strategy called fine tuning, where some parameters in the net are frozen and the rest of the parameters are retrained with gradient descent and backpropagation (Gao & Mosalam, 2018). This is to avoid overfitting problems and this method is less time consuming compared to regular CNN training from scratch (Gao & Mosalam, 2018).

Convolutional neural networks (CNN)

According to Indolia et al.(2018), CNN is described as the concept of hierarchical feature detectors in a biologically inspired manner. It remains one of the renowned deep learning architectures aimed at image classification and object recognition (Jafar & Lee, 2021). CNN’s architecture follows the idea of shortcut connection that skips layers which in turn reduces training time and overfitting the model (Jafar & Lee, 2021). According to Albawi and Mohammed (2017), convolutional neural networks had great results in a variety of fields and most importantly it reduced the number of parameters in ANN. Besides, Albawi and Mohammed (2017) also mentioned that problems that are solved by CNN should not be spatially dependent. Acharya et al. (2017) say that the accuracy of CNN by changing from augmented data to highly imbalanced data (original dataset) decreased only by 5 per cent and when the model is properly trained, CNN can produce a fairly accurate result. Rezende et al (2017) also stated that ResNet-50 performed better than the average GIST feature extractor proposed in the literature.

Residual Network (ResNet)

Residual Networks, also known as ResNet is a type of network which works by subtracting features learnt from layer inputs (Amaya-Rodriguez et al., 2021). Its idea stems from accumulating identity mappings and through skipping unimportant layers in the network whilst also reusing activations from trained layers (Jafar & Lee, 2021). As a result, Resnet's ultra deep networks do not suffer from vanishing gradient problem (Alom et al., 2018). The strategy of a shortening edge in ResNet allows the creation of deeper networks as many as 150 layers which was not possible before (Bhardwaj et al., 2018). According to Amaya-Rodriguez et al. (2021), with this many layers, ResNet still achieves compelling performance. Different depths of ResNets have different purposes. Using a deeper pre-trained ResNet model for large datasets allows the excavation of more abundant deep information, which is more well suited for feature extraction (Liu et al., 2019).

3.6. Summary

In conclusion, this brief literature review has covered the ongoing issues present in medical data such as social, ethical and heterogeneity issues. The idea of transfer learning aims to increase a model's accuracy by reducing the chances of overfitting the model as opposed to building a raw model from scratch (Gao & Mosalam, 2018). CNNs produce a better accuracy when the model is properly trained (Acharya et al., 2017). Residual Networks have also risen in popularity for image detection as it skips layers which reduces training time (Jafar & Lee, 2021).

4. Outcomes

4.1. What has been implemented

Over the course of the project timeline, our team has adopted implementations in line with what was previously mentioned in our project proposal, except for some deviations on a few technical aspects of our project. For our predictive model component, the given dataset is split into a 6:2:2 ratio divided into training, validation and testing datasets. Then, data augmentation and normalization such as random resized crop to the designated size (according to ImageNet standards), random rotations and horizontal flips, colour jitter, and normalization is done according to mean and standard deviations were applied to our given image dataset. Our team then implemented a DataLoader, according to PyTorch standards with a batch size of 16. For our model, our team has chosen to train on a pre-trained residual network from the package resnet50, which is a deep learning algorithm based on Convolutional Neural Networks. The transfer learning techniques that were applied to our trained model involves back propping to every parameter. This means that we are finetuning the Convolutional Neural Network as opposed to feature extraction. The classifier architecture is implemented to ensure that the size of each output sample is constant with the two different classes of our dataset - MSS and MSIMUT. The loss function specified for the model is the CrossEntropyLoss function, whereas the optimization function uses Adam optimizer. Lastly, a scheduler is also implemented, where the LR is decayed by a factor of 0.1 every 7 epochs. The resulting model is then trained on a given number of epochs, where the best model is then saved and fed into both the inference notebook and the integration to the website. For the inference notebook, the model is fed and set into evaluation mode to view its confusion matrix and AUC scores.

For our website, the backend logic is created using a Python framework called Flask. Flask's infrastructure allowed us to build a web application with various tools and libraries. To maintain consistency with previous plans mentioned in the project proposal, we have implemented several functions linking to the designated page on our website such as About, Help, Sign up, Login buttons. The signup and login functionalities were implemented using a database package from Flask called SQLAlchemy. To maintain users' privacy, our team has also utilized a secure hash function to encrypt our users' passwords from a package called werkzeug.security. The previously trained model is then fed into our website code, and an additional function called predict would output the user's classification (MSS or MSIMUT) and its percentage of contracting it.

4.2. Results achieved / Products delivered

Most of the results achieved based on the implementation described above have produced outcomes that were expected. For our predictive model, training using a pre-trained resnet50 in addition to the combination of batch size, optimizer, loss function, LR scheduler and the thirty-five epochs mentioned previously have allowed us to develop a model with a decent accuracy of 81.4%. Our team has also delivered graphs as these serve as an additional tool to measure the effectiveness and efficiency of our model. These graphs include the training loss and validation loss against each epoch during training shown in Appendix A, the training accuracy and validation accuracy against each epoch during training shown in Appendix B, and lastly, the confusion matrix and area under the curve (AUC) (shown in Appendix C) generated from our trained model during evaluation.

The successful integration of our predictive model onto our website component and succeeding deployment on Heroku's infrastructure meant that we were able to achieve both our minimum viable product (MVP) and one of our primary goals of this project to ensure its success. The successful result stemmed from this would mean that users can choose an image to upload and predict their risk of contracting cancer, including additional functionalities such as creating an account.

Extending from the results obtained above, this would allow us to successfully deliver both the actual website and documentation - both technical and non-technical from our workflow. The actual website can be accessed from https://cancerprediction-4.herokuapp.com/ , whereas technical documents including the main and inference Jupyter Notebook containing our code for the predictive model component and main python script for our website would also be included in our deliverables. Non-technical documentation such as User Guides and Test Reports, in addition to this report, would also be provided in extensive detail by our team.

4.3. How are requirements met

As previously mentioned in our project proposal, to meet the requirements expected from our stakeholders, our team drew an extension from our preliminary requirements stemming from our project to create user stories that would serve as a baseline for suitable acceptance criteria. In addition to this, our team has also maintained a log of our requirements traceability matrix, shown in Appendix D and made a few adjustments to it based on deviations made to our project requirements. An updated version of our requirements traceability matrix can be found in Appendix. One of the user stories that has successfully been implemented into our project workflow is to let users sign up, log in and predict their risk of contracting gastrointestinal cancer. Another acceptance criteria that was maintained relates to the integrity and handling of users' private information such as their passwords when logging in to their accounts, which we have accomplished during the development of our website component of our project.

4.4. Justification of Decisions Made

To reach the current stage and timeline of the project came from a series of discussions made by our team members over the course of the project. As these decisions were pivotal in affecting our project's successes and failures, our team members made sure to weigh out the pros and cons of each methodology before committing to implementing them. Firstly, for our predictive model component, our team has chosen the PyTorch infrastructure as opposed to Keras mainly due to us as team members not having any prior experience with deep-learning mechanisms and methodologies and PyTorch serves as the basic stepping-stone for beginners to implement deep-learning models as its simpler to code and easier to understand (Kothekar, 2021). Our team has also used the pre-trained package, resnet50 which is a residual-learning algorithm that consists of 50 layers as it has a shorter training time compared to other modules and reduces the risk of overfitting the trained model as it trains on each layer residually (Narayanan, 2019). The transfer learning methodologies implemented on our predictive model fine-tune the weights of the pre-trained network by continuing the backpropagation, which as the layers go deeper, can provide a more detailed classification of the classes, where we hope that accuracies resulting from this would be higher as opposed to fixed feature extraction (Stanford University, 2021).

Our team also chose Adaptive Moment Estimation (Adam) optimizer as opposed to conventional Stochastic Gradient Descent (SGD) due to Adam optimizers being able to adapt per-parameter change in gradient descent and its overall effectiveness in achieving a better-trained accuracy model during the rigorous training phase of our project (Brownlee, 2021). The Cross-Entropy Loss function was also chosen as opposed to the sum of squares due to its shorter training time and its improved generalization over classification problems by minimizing the distance between two probability distributions - predicted and actual (Narang, 2019; Brownlee, 2020). As the given dataset consists of only two classes - MSS and MSIMUT, it becomes a binary classification problem which ultimately makes it a good match for our training of our predictive model (Brownlee, 2020). For our batch size, our team paid extra attention as it is one of the most important hyperparameters to tune in modern deep learning systems. We decided on using a batch size of 16 as from manual trial and error, trained accuracies tend to be higher on lower batch sizes. Larger batch sizes increase granularity and use much more memory and GPU power (Shen, 2018).

4.5. Discussion of All Results

As previously mentioned, the various results stemming from the above implementation has allowed us to achieve our minimum viable product (MVP) that is to have a presence of a trained and working predictive model, with an accuracy of at least 60% integrated onto the website infrastructure developed using Flask without database functionality. However, the actual outcome leading up to our delivered product achieved better results. The final trained predictive model integrated onto our website had an 81.4% accuracy, and database functionality linking to sign up and login pages on our website have also been successfully developed and deployed. This means that through the implementation described above, our project outcomes have been largely achieved, successful and kept up to date with stakeholders' expectations and requirements with no major bugs present on our product and extensive room for improvement in the future. The choices our team made on the parameters such as optimizers and schedulers have also proved to be largely successful in aiding for smooth integration between the predictive model component and website component of our project. Overall, the results that our team members achieved has been largely in line with the expected outcome that was projected and mentioned in the previous project proposal.

4.6. Limitations of Project Outcomes

Despite our team achieving results that were largely expected, the project implementation carried out by our team members had some flaws in them, especially in cases where the results achieved were sufficient to fulfil the MVP but lacked in the original description mentioned in the project proposal. One of these cases relates to the accuracy of our developed model. Initially, the projected accuracy was supposed to be ~95%, but during the execution of the project, it turned out to be unrealistic, as we were unable to find the correlating parameters that would give out a better accuracy. In addition to this, the given dataset was very limited in size, which meant that we could only work with what was given to us, and couldn't take into account any other external data sources. In terms of technical limitations, our team members were largely bound by long hours of training time, and the lack of resources available to train with other pre-trained models such as AlexNet, DenseNet, etc. For our website functionality, one of the limitations was in our program not being able to show the image uploaded by our users. Although this limitation does not pose a threat to the integrity of our program, it was mentioned in our proposal, so our team put our efforts into fixing the bug, but unfortunately to no avail. Since our website's upload functionality is still in its preliminary stages, it can't differentiate between medical and non-medical images which lead to our website handing out inaccurate results to our end-users that can prove to be fatal.

4.7. Discussion of Possible Improvements and Future Works

As an extension of our limitations discussed above, our team recognizes that there is room for improvements to be made on future developments on this project. If given the chance to expand our team environment, there is potential to develop a better website that would take into consideration other factors such as form questions to detect their risk of contracting gastrointestinal cancer, instead of just uploading images. The predictive model could also be further developed if given a larger dataset that comprises more MSS and MSIMUT images, and through the increase in resources, could lead to training from different packages such as DenseNet at a shorter time with hopefully, a better accuracy. Another way to better increase the accuracy of our model is to re-train our model with the uploaded images from our users, as these images are essentially non-dataset images that our model has not seen yet. On the website side, enhancements can be made to the upload feature to differentiate medical and non-medical images. Time resource is also an important factor, as greater amounts of time lead to greater amounts of improvement that could be made into our end-product. For future developments, our team can look into incorporating into health institutions' API that can cover varying types of cancer instead of just gastrointestinal cancer.

4.8. Summary of Test Report

To ensure that our delivered software is up to the quality standards, extensive testing approaches of Black Box, White Box, and Usability testing were done within bounds of possibility from our software. During testing, several flaws were identified from our website that required further integration testing to rectify. For example, in our upload image functionality, previously our algorithm was not able to differentiate between medical and non-medical images. As a result, an exception handling was created in attempts to catch these false inputs, albeit not being completely perfect. The testing approaches were also limited in variability, and our team did not implement automated and regression testing, due to the limited functionalities of our website and lack of expertise from our team members. This reduced the chance for our team to further enhance the developed model that can help in improving its accuracy. Limitations with regards to the security of our software were also exposed, as our team realized the low levels of encryption done to protect users' sensitive information. Overall, the testing approaches have helped improve the robustness of our software, but more extensive testing is needed to ensure that our developed software is up to industry standards.

4.9. Critical Discussion on Outcome

Overall, the implementations done over the course of the project timeline have proved to be largely successful. Our team executed the project by first utilizing transfer learning techniques on a pre-trained Convolutional Neural Network from the package resnet50 and altering the required parameters such as optimizers, loss functions and schedulers. This is then fed onto our website infrastructure and deployed so our end-users can access and view their results. As with all project environments, the results that our team achieved were not perfect and there were some room available for improvement especially on the resource and time management side of the project where our team discussed that if given a longer time to execute the project, the quality of our end-product delivered would have been better.

5. Methodology

5.1. Design

The external design aspect of our project was mostly finalized in the first half of the project. During the implementation stage, our team tried to incorporate the initial design of our website user interface (shown in Appendix E), however, some alterations in non-technical features such as background colour, font sizes were made (shown in Appendix F). All of the technical functionalities present on the navigation bar - About, Help, Sign up and Login pages on the preliminary design were carried over to the actual implementation stage without any change in its designated functionality. As mentioned previously, our website component is developed using Flask, accompanied by raw HTML and CSS for the various pages and styles present on our website. Our team was unable to incorporate the actual colour scheme and background as we opted to use a pre-designed template taken from OS Templates.

An updated overall architecture of our software design linking both the front-end and back-end side of our website can be seen in Appendix G. This updated version provides a high-level task overview at each stage of our software infrastructure highlighting what our team has implemented over the course of the project.

5.2. Implementation

5.2.1. Tools

<u>Programming Language</u>

The programming language that the team has chosen for this project is Python. This language is used for both main components of the software – predictive model and website. HTML and CSS are used for the front-end of our website.

<u>Libraries and Frameworks</u>

To fulfil the requirements of the software that our team is building, libraries and frameworks have been chosen to ensure the development process runs smoothly. For the predictive model, the libraries NumPy, pandas, matplotlib, and PyTorch have been chosen. These libraries allow us to develop the model by allowing us to process data, create informative visualizations, as well as train our model. For the website, the micro-framework – Flask – is chosen to be able to build a simple web application. To help with the database system of our website, we are using the SQLAlchemy library. To create a user authentication system, flask_login is used and to maintain the safety of users' passwords, werkzeug.security is used.

5.2.2. Softwares and Platforms

To support the development of the software that our team is creating, we have chosen various software to implement and store our codes. Implementations of both our predictive model and Website code are done in Visual Studio Code using built-in extensions for Python and Jupyter Notebook. For storing local databases, our team opted for PostgreSQL, in line with website deployment on Heroku. To maintain version control in our project, GitHub Desktop is used.

5.2.3. Hardware

The hardware used to code our Jupyter Notebook and main python scripts were done mostly on our devices, which comprises both Mac and Windows Operating systems. Training of the predictive model was done on a Windows Device with GPU Compatibility (Nvidia GeForce RTX 2070 Super).

5.2.4. Algorithm

As mentioned above, our developed predictive model utilizes transfer learning on Convolutional Neural Networks from the package resnet50. This is used to classify the dataset images to determine MSS or MSIMUT. Parameters affecting the accuracy of the model such as Optimizers and Loss functions are fine-tuned manually during each training session. Each epoch in our training function comprises a training and validation epoch.

5.3. Justification of Decisions Made

Our team chose the programming language Python, due to all team members familiarity with the language, sufficient background knowledge, simple syntax and ease of use. For our source code editor, Visual Studio Code is used due to its speed and its features such as syntax highlighting, bracket-matching etc. As the project environment was remote, the version control system played an important role in our project. In our case, we used GitHub due to its easy configuration and seamless collaboration without compromising the integrity of the original project.

For our website component, we implemented the Flask architecture as it is ideal for small team environments and leaves room for further developments to be made. Its simple and lightweight design makes it easy for us to code, as none of us has had any prior experience in building a website before (Holcombe, 2020; Mieczkowski, 2021).

6. Software Deliverables

6.1. Summary of Software Deliverables

6.1.1. What is Delivered

The project deliverables in this project can be divided into two categories – project management-related deliverables and product-related deliverables.

The project-management related deliverables include a business case along with a weighted scoring model. A project scope statement is also delivered to describe our software requirements and characteristics. A requirements traceability matrix is also included to further explain the functional and non-functional requirements of the project. Also included in the project management-related deliverables is a risk register to keep track of potential risks in the future and resolve risks in our project.

The main product-related deliverable is the website where users will be able to predict the chances of contracting gastrointestinal cancer using the predictive model that has been integrated into the website. We have included the software code for the predictive model and the website itself. Along with the website, we have also included an end-user guide to guide users through the steps from how to sign up / login, uploading their image to predicting their risk. Additionally, a technical user guide is also included as an extensive guide for the system administrators to be able to set up and configure the software we deliver. The technical guide will explain how to set up the predictive model and the website. Further deliverables include a research report that covers our research on cancer predictive modelling. Additionally, a demo is also given to explain how to access our website and use its functions. Lastly, a software test report is included to show how our software meets the requirements, how different conditions and behaviours are handled in the software, limitations on the software, as well as recommendations for future improvements.

To sum up, our project deliverables are as follows:

Project management-related deliverables:

1. Business Case
2. Weighted Scoring Model
3. Scope statement
4. Requirements Traceability Matrix
5. Risk Register
6. Final Project Presentation

7. Final Project Report

Product-related deliverables:

1. Predictive model software code
2. Research Report
3. Website for cancer prediction
4. Website design document
5. Website software code
6. End-user and technical guide
7. Code Demonstration
8. Software Test / QA Report

6.1.2. Sample Screenshots and Description of Usage

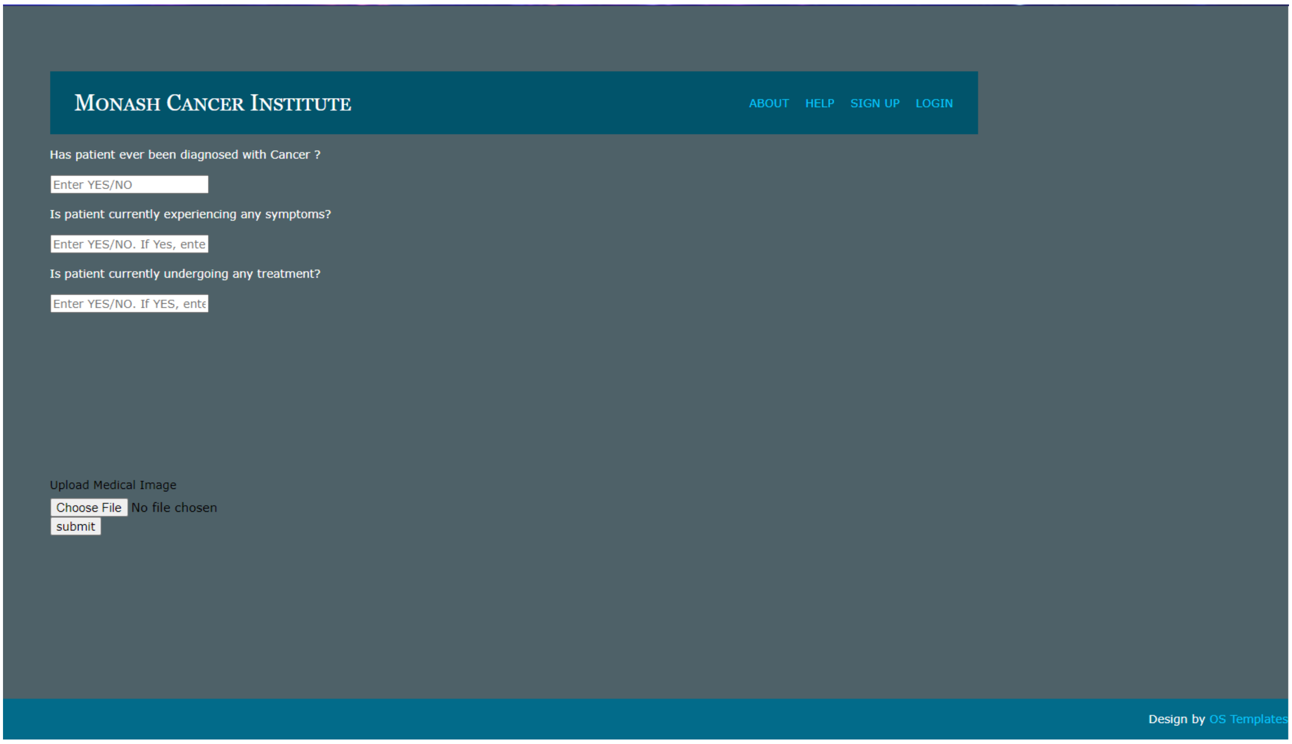


Figure 1 - Landing Page of Website (Ooi et al., 2021)

Figure 1 shows the page loaded when the users first access the website (https://cancerprediction-4.herokuapp.com/). On this page, users will be able to sign up, log in, and upload images to predict the risk of contracting gastrointestinal cancer. Also on this page, the users can access the about page as well as the help page in case users need more guides to be able to use the website.

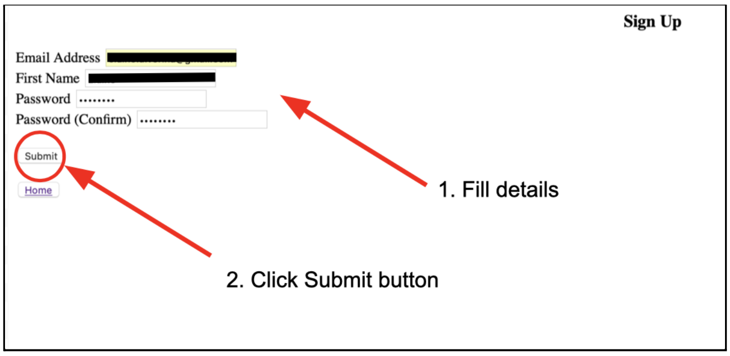


Figure 2 - Sign up Form (Ooi et al., 2021)

When a user wants to sign up, they will be directed to the page as shown in Figure 2 to fill in their email address, first name, as well as password.

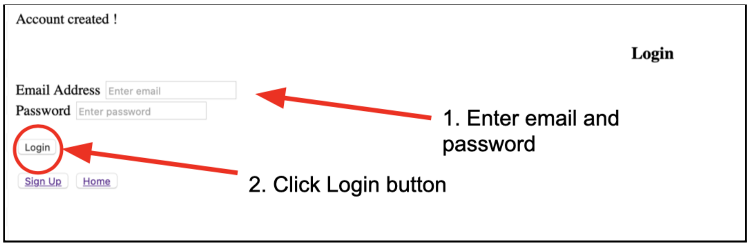


Figure 3 - Login Page (Ooi et al., 2021)

After the account is created, the users will be directed to log in as shown in Figure 3. After that, users will be brought back to the main page where they can fill up some questions and upload an image to predict.

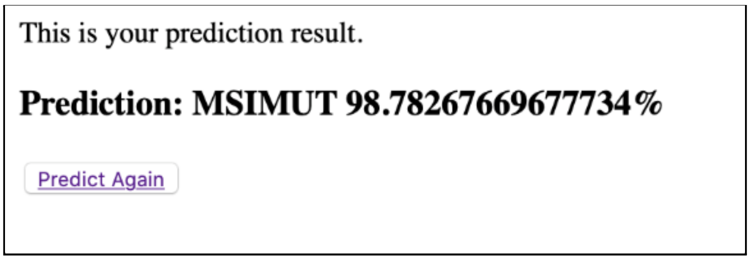


Figure 4 - Prediction Result (Ooi et al., 2021)

Lastly, after a user submits an image and predicts, a page similar to Figure 4 will be shown. Here, the user can see their uploaded image, and see their prediction result as the probability of having an MSS or MSIMUT result. The user will also be able to click the Predict Again button to go back to the main page, where they will be able to upload another image to predict.

6.2. Software Qualities

6.2.1. Robustness

Several methods have been used to maintain the robustness of this software. Firstly, before the submitted file is processed, the software will test for the existence of the file. Secondly, the file formats that the users submit have been restricted into 3 types which are jpg, jpeg and png. If the rule is violated it will direct the user to an error file. However, our software is not able to validate the input data of the submitted file. For example, if the users submit a completely irrelevant image or submit a similar medical-like image, the software will still process and produce a result. To counter this, an exception handling has been created to handle some irrelevant inputs. Furthermore, for the user authentication system, all input data will be validated, and if the input data is in the wrong format, it will produce a flash message and show it to the users.

6.2.2. Security

User Authentication

User authentication systems are achieved by SQLAlchemy and Flask-login modules. The system ensures the user is whom they claim to be by using the flask-login function called check_password_hash. The users' passwords are kept safely with a hash function from werkzeug.security modules. However, the users' first names and email have a risk of a data breach as no hash function is used to protect this information. Two-factor authentication is not available in our project.

User Authorization

Authenticated users are allowed to carry out what they are trying to do with the help of the flask-login function called current_user.is_authenticated. This allows the system to check its identity before executing relevant code. There is only a single user level in our system. However, only logged-in users can save their responses to the database.

Access

As our online tool deals with highly sensitive data, our user's integrity and private information must be protected. Although the encryption methods implemented on our project can be seen as preliminary, only the software developers - particularly the developer that deployed the website on Heroku's infrastructure has access to the contents of the database. This reduces the chances of a data breach happening at a larger scale as only the developers are at risk.

6.2.3. Usability

For the usability of our software, our website layout is user friendly and its colour has good contrast with a mix of light blue and dark blue. All the buttons are well positioned and in white colour so it allows users to quickly access the button. Furthermore, there are example inputs in the responses for the users to know what to enter. Moreover, there are multiple buttons on all the pages so the users can navigate throughout the website easily. There is also a help button on the top right of the homepage to provide step-by-step instructions for the users on how to use our website. Lastly, error messages will be shown if the users provide the wrong format inputs.

6.2.4. Scalability

As our project environment is bounded by university constraints, it leaves us with no budget to make extensions to our website. Currently, our website is deployed on Heroku's infrastructure, making use of Heroku's free memory usage of up to 512 MB on free accounts. Our website presently occupies around 315 MB of usage, which slightly exceeds the soft slug size on Heroku's infrastructure that can slightly affect the boot times of our website. However, this doesn't mean that there is no room for expansion on the scalability of our project. The flexibility offered by our infrastructure means that in future cases where resources are highly available for us, it becomes viable to scale our website to manage greater amounts of traffic and have a wide range of functionalities.

6.2.5. Documentation and Maintainability

All of our codes are well documented so the maintainability of our code is ensured. There is documentation on most of the functions on what the code does, the expected input and output and unusual conditions. Besides, our code has easy readability with good usage of white spaces and indentation. All related codes are close to each other and all the routes are mentioned properly. No inconsistent naming of variables and all variables and functions have meaningful names. These will make it easy for modifications and updates in the future, especially if different technical users need to make those changes.

6.3. Sample Source Code

A sample source code of our software is included in Appendix H. This sample code serves as a representative of our team's overall source code. The first set of code shows how the user-submitted images are submitted and how the submitted images are processed with two functions: transform_image(image_bytes) and predict(image) and display the result on result.html. This set of code shows how one of our software's main functionality, which is to predict an image's risk of contracting gastrointestinal cancer, is being satisfied.

7. Critical Discussion on Software Project

7.1. Execution

Throughout the project timeline, our team has experienced ups and downs in both expected and unexpected scenarios. Although our team did manage to get the outcomes that were expected, the overall project execution experienced by our team was sloppy and poor, especially during the early stages of the second half of our project. One of the major contributions to this flaw was the lack of high-level research done after the completion of our project proposal leading up to the earlier weeks of the second half of the project. This also meant that our team did not follow through with the schedule mentioned in the Work Breakdown Schedule and Kanban Boards, which resulted in a few risks from our risk register being triggered (Appendix I). The outcome of this poor execution meant that our team struggled with bottlenecks resulting in poor work efficiency. For example, our team had not conducted prior research on the software applications that we were going to use. Initially, we had used PyCharm IDE for our predictive model, then we came to realise, in the middle of our progress, that we needed a Jupyter Notebook instead. This resulted in the redundancy of repeated work as we had to transfer our work onto a different software application. It was also clear that our team lacked forward thinking due to the lack of research and initiative from each of our team members to complete the project. In addition to this, none of our team members had any prior experience in coding/building a predictive model which added to our pains and frustration as we could only manage little progress at each weekly meeting.

Through the perseverance and determination of our team members, the level of execution became better towards the middle of the project timeline, and our team members slowly got back into the rhythm and went in line with the schedule initially proposed. During this time, our team members learnt the most, in both the technical and managerial aspects of our project. As we progressed through each delegated task, we were able to cross-check and discuss each methodologies' pros and cons in making a team decision, which has greatly helped us in making better decisions that would in turn result in a better product delivered leading to higher satisfaction from our stakeholders.

7.2. Deviation

While the overall outcome of the project went well, several components in the software required some deviation from the initial plan that we proposed. For example, the development of the website and the package we used to train the predictive model.

In the initial project proposal, we have mentioned that we would be using an application called WordPress to build the website. However, due to unfamiliarity with the application and some feedback from the project supervisor, we decided that it would be more appropriate for our team to use raw HTML and CSS for the overall layout and style of our website using a template from OS Templates. In addition, the logic behind the website is implemented using Python, utilizing a framework called Flask. Furthermore, we have also implemented a database system for storing users' login information using SQLAlchemy and PostgreSQL instead of utilizing MySQL as mentioned in the initial proposal. This is also due to the unfamiliarity with MySQL.

We have initially proposed that we would use the resnet18 package for the development of the cancer predictive model. However, our group decided to use resnet50 over resnet18 after some trial and error in creating the model. We were able to achieve a higher accuracy model using resnet50 (81.4% accuracy) compared to using resnet18 (68.9% accuracy).

As the satisfaction of our stakeholders is our utmost priority, our team decided that these deviations from our initial project proposal are necessary to achieve a product that not only meets the requirements of our stakeholders but also exceeds their expectations on the final product.

7.3. Failures

While our project outcome can be largely considered successful, there are a few components of the software where our team failed to meet the initial proposal's requirements and expectations.

For our predictive model component, one of the technicalities introduced in our proposal was to develop a trained model that had an accuracy of greater than 95%. However, given the constraints experienced by our team members, we were unable to achieve this. This was largely because our project environment was bounded by the given dataset and the lack of other pre-trained models such as DenseNet and Alexnet tried on to obtain a better accuracy model. Due to the lack of high-end specifications hardware, the training time posed by our team became a bottleneck, meaning that time was a consideration as our team needed to deliver a successful trained predictive model. To add to this pain, our team was largely unfamiliar with instantiating a Virtual Machine Instance on Google Cloud servers and coupled with the poor internet connection posed by our team members, the alternative solution was not viable and soon fell apart.

For our website component, our team was largely unaware of the software qualities such as robustness, security and scalability which meant that the current quality of our website is still in its infancy state. Our team considers the given execution a failure since our project deals with highly sensitive personal information, including medical images that pose a threat to the integrity of our website that could further lead to lawsuits regarding the breach of personal data from our end-users. Our team members should have paid more attention to the development of these software qualities to deliver a better product for our stakeholders.

The failures described above have been a stepping stone for the team members present in this project and if given the opportunity in the future to recommence the project, these are the qualities that our team members would pay more attention to and reduce the instances of these failures re-appearing.

8. Conclusion

To sum up, this report has extensively covered our team's progress in attempts to successfully create software that predicts the early stages of gastrointestinal cancer using a predictive model, utilizing deep learning methodologies, that is integrated into a website. This software will be publicly accessible for health institutions to make use of. Our team aims to train a CNN model which has a high accuracy – ideally greater than 95%.

When delivering our product, our team prioritizes the satisfaction of our stakeholders. Hence, our stakeholders' feedback during the span of the project is immensely valuable to ensure all requirements are delivered. As our team is using the Agile methodology, working with short cycles – one week – made it possible to continuously receive meaningful feedback from our stakeholders to help shape our decision-making process throughout the project.

The implementation process of our software mostly follows what has been proposed earlier in our project proposal. Firstly, for the predictive model component, conventional data augmentation and normalization are applied to the dataset, and parameters such as Optimizers, Loss Functions, Schedulers and Epoch Size are manually altered and trained to obtain better accuracy. For the website component, our team uses Flask for the backend logic. Following the initial design, our team has implemented functionalities such as a navigation bar as well as sign up and login functions. Additionally, to maintain users' privacy, our team uses werkzeug.security package to encrypt users' passwords. Lastly, the predictive model trained earlier is integrated with the website and deployed on Heroku's infrastructure. These results that we achieved have exceeded our minimum viable product (MVP) requirements.

Most of the design aspects of our user interface remained consistent with the preliminary design made earlier in the year. Alterations were only made to the typeface and colour of our website. In addition, our team has delivered project-management-related and product-related deliverables. They include a business case, weighted scoring model, scope statement, requirements traceability matrix, risk register, final project presentation and report, predictive model software code, research report, the website for cancer prediction, website design document, website software code, end-user and technical guide, code demonstration, as well as software test / QA report. All deliverables can be found in the appendix of this document or on a separate document.

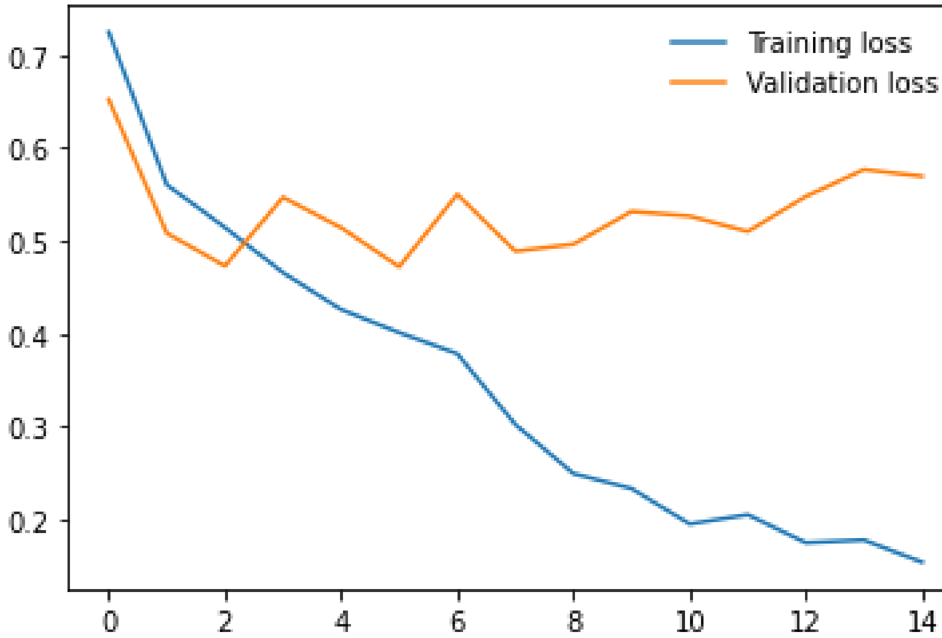
To ensure the high quality of our software, we have ensured the expectations on robustness, security, usability, scalability, documentation and maintainability are met. Preliminary attempts to increase the security and usability of our software were made. In regards to scalability, currently, due to budget constraints, it is not possible to make extensions. However, expansion in the future is possible due to the flexibility of our infrastructure. Lastly, for the documentation and maintainability aspect, we have ensured that our codes are well documented with comments of the expected input and output at every function. The code implemented uses proper white spaces and indentation for readability and variable names are consistent for easy modifications in the future.

This report also covered the team's critical discussion on the executions, deviations and failures of the project. For example, during the execution of the project, our team struggled immensely with time management and was not on track with the designated schedule. Some deviations were also made during this project such as altering the pre-trained model from resnet18 to resnet50 and changing from WordPress infrastructure to Flask infrastructure. In terms of failure, the methodologies implemented by our team had flaws caused by both our unfamiliarity with the overall procedures of our project and not fulfilling the criteria mentioned in our initial project proposal, such as our final predictive model accuracy. Our team was unable to fully develop the software qualities such as robustness, security, and scalability. Our team should have given more attention to these qualities as we are dealing with highly sensitive information. These are the aspects of the project where we could improve if allowed to work on this project again in the future.

9. Appendix

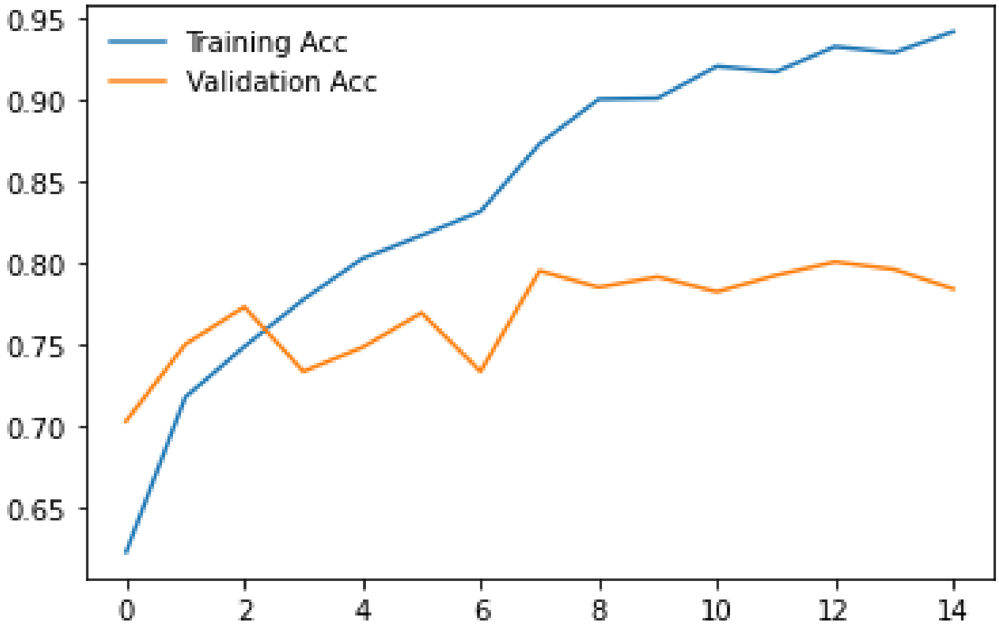
Appendix A

Predictive Model Loss



Appendix B

Predictive Model Accuracy



Appendix C

Confusion Matrix and AUC Score of Best Model

```
Confusion matrix for resnet50:
[[299 135]
 [ 88 564]]
AUC score for model resnet50: 0.7769853835062621
```

Appendix D

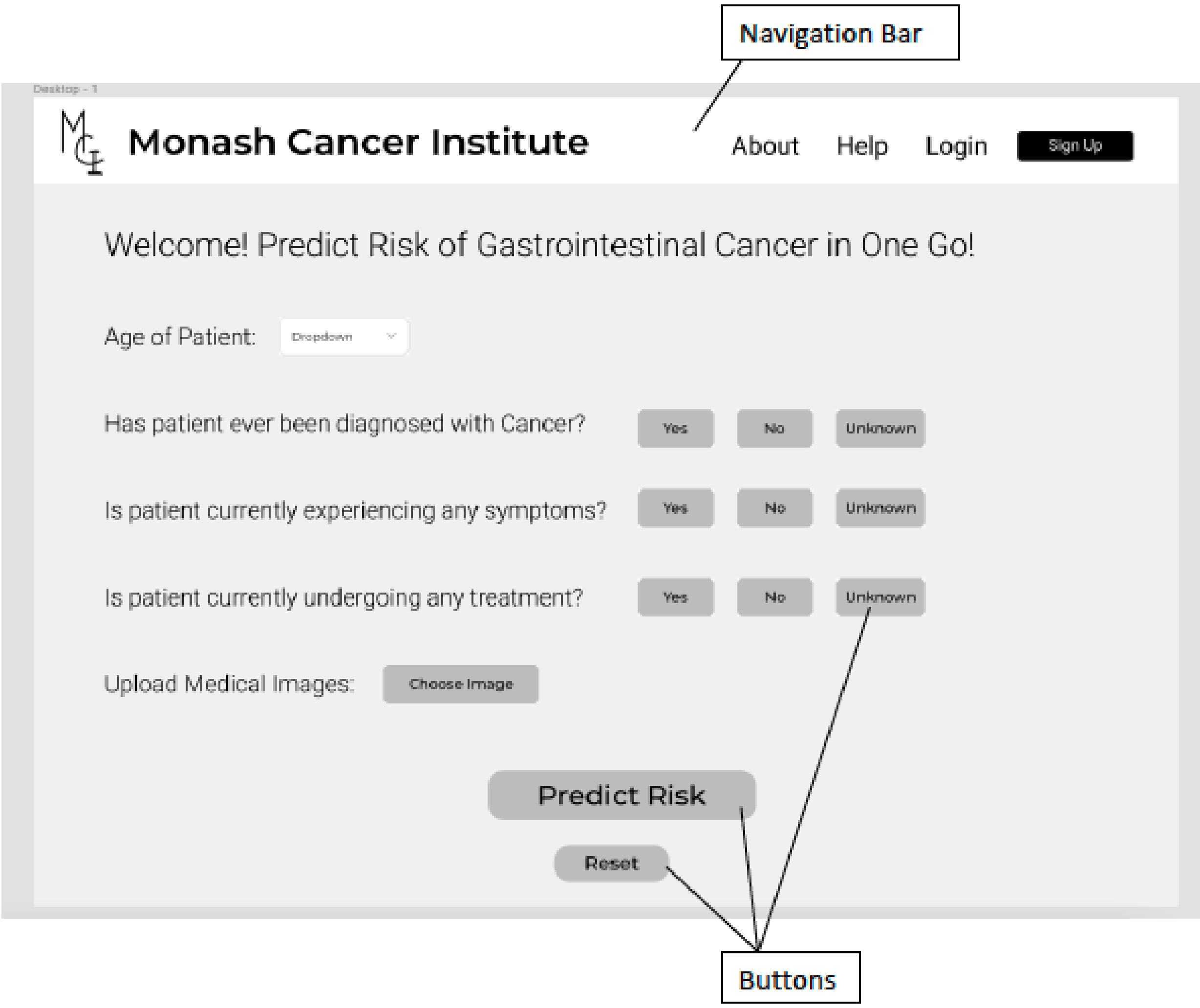
Requirements Traceability Matrix

REQUIREMENTS TRACEABILITY MATRIX					
Project Name:	**Data Mining Technique To Detect Cancer Using Predictive Modelling**				
Project Manager Name:	**Vionnie Tan**				
Project Description:	**Building a predictive model to determine early stages of cancer**				
ID	***Requirements (Functional or Non-Functional)***	***Assumption(s) and/or Customer Need(s)***	***Category***	***Source***	**Status**
001	Image Processing - Splitting Dataset into 6:2:2 ratio, and Data Augmentation and Normalization	Source of images are come from a legit source	Functional	Kaggle	Completed
002	Programming Skills - Understanding topics regarding AI, Machine Learning, Deep Learning, Transfer Learning	Extensive knowledge of these programming skills increases the chance of efficiency of our predictive model	Functional	Online resources such as Stack Overflow, Leetcode, and Monash Units	Completed
003	User Friendly Interface to allow login and signup from our users	Our stakeholders have to be able to easily access and understand the interface for them to use the website	Non-functional	Project supervisor	Completed
004	Stakeholder expectations met	Accuracy of the Model has to be > 60% (MVP)	Non-functional	Stakeholders	Completed
005	Predictive Model developed is integrated into Website code	Website code should be able to access the trained model and integrate it onto the	Functional	Stakeholders	Completed

		website			
006	Identify important predictors that have significant impact on successful cancer categorization, through transfer learning	Training on every layer in the resnet50 model as opposed to feature extracting	Functional	Project supervisor	Completed
Documentation: (Include any justification and assumptions made)					

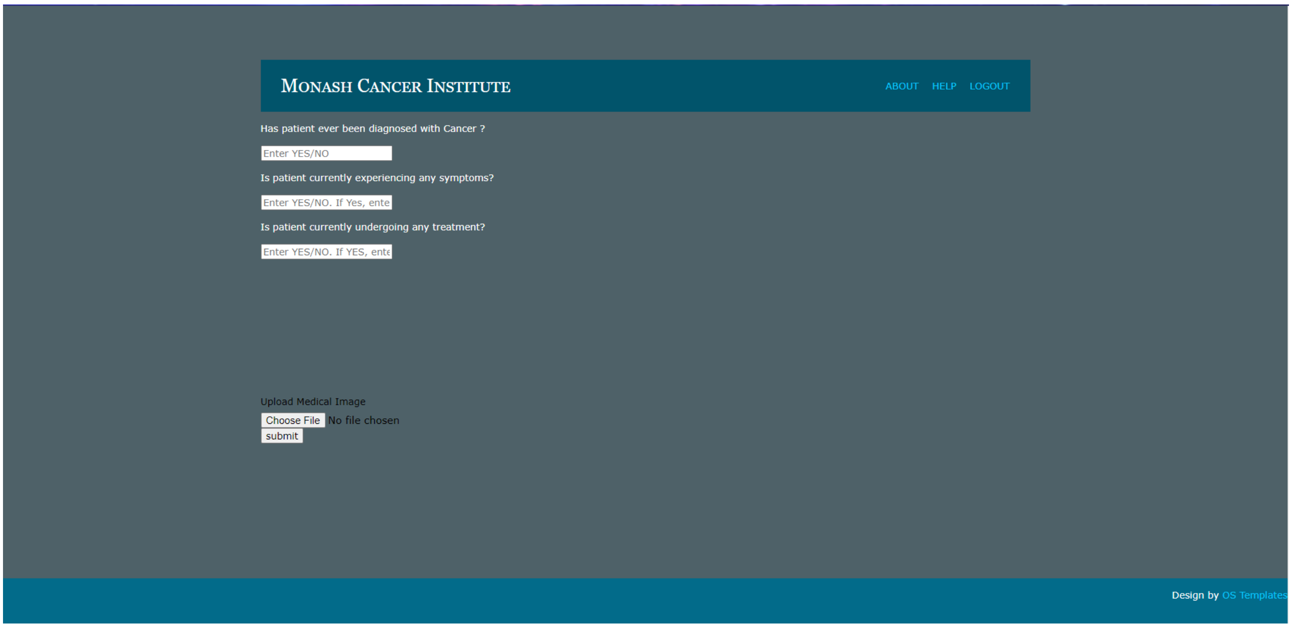
Appendix E

Initial Website Design (Landing Page)



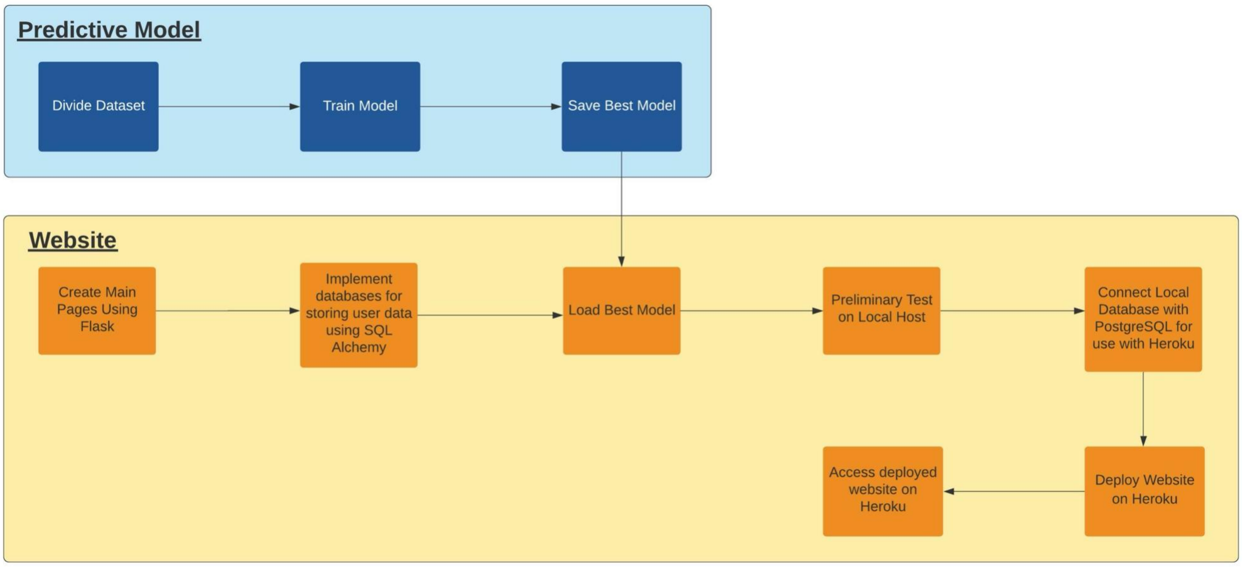
Appendix F

Final Website Design (Landing Page)



Appendix G

Software Architecture



Appendix H

Sample Source Code

First set of code:

```
@app.route("/", methods = ['GET','POST'])
def home():
    """
    Route of homepage, display the homepage to the user and listen to GET
and POST, after user submitted image, run through the predictive model
    @special condition: sometimes will produce result if the submitted
images are unrelevant
    @expected output: output the result of predictive model
    @return: render the homepage HTML, if user submitted image, render the
result html with the prediction result
    """
    error = None

    if request.method == "POST":
        # when user submit image
        if request.form["submit"] == "submit":
            vCancer = request.form.get('vCancer')
            vSymptoms = request.form.get('vSymptoms')
            vTreatment = request.form.get('vTreatment')

            if current_user.is_authenticated:
                update_user = User.query.filter_by(email=
current_user.email).first()
                update_user.vCancer = vCancer
                update_user.vSymptoms = vSymptoms
                update_user.vTreatment = vTreatment
                db.session.commit()

            #check if the post request has the file
            if not request.files.get('file',None):
                return render_template("error_empty.html")
            file = request.files.get('file')

            #if whether the submitted image are in jpg, jpeg and png format
            if ("." in file.filename and file.filename.rsplit(".",
1)[1].lower()) not in ["jpg","jpeg","png"]:
                return render_template("error.html")

            if not file:
                return
```

```
            try:
            #run the predictive model with the submitted image
                img_bytes = file.read()
                prediction_name, percentage = predict(img_bytes)

            except:
                return render_template("error_file.html")

            #add users responses to the database
            if current_user.is_authenticated:
                update_user = User.query.filter_by(email=
current_user.email).first()
                update_user.result = percentage
                db.session.commit()

        return render_template("result.html", name= prediction_name,
prediction = percentage)
    return render_template("index.html", user = current_user)
```

```
model = models.resnet50()
num_inftr = model.fc.in_features
model.fc = nn.Sequential(
    nn.Linear(num_inftr, 256),
    nn.ReLU(),
    nn.Dropout(0.4),
    nn.Linear(256, 10),
    nn.LogSoftmax(dim=1)
)
model.load_state_dict(torch.load('model.pth',map_location=torch.device('cpu'
)))
model.eval()

imagenet_class_index = ['MSIMUT', 'MSS']

# SOURCE:
https://towardsdatascience.com/build-a-web-application-for-predicting-apple-
leaf-diseases-using-pytorch-and-flask-413f9fa9276a
# Pre-process image
def transform_image(image_bytes):
    """
    This function performs image transformations on an image
    :param image_bytes: an image
```

```
    :return: image_bytes that has been transformed as the following:
            1. resized to size 256
            2. cropped at the center with size 224
            3. converted into tensor format
            4. normalized with mean and standard deviation
    """
    my_transforms = transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])

    image = Image.open(io.BytesIO(image_bytes))
    return my_transforms(image).unsqueeze(0)

def predict(image_bytes):
    """
    This function predicts whether an image is classified as MSS/MSIMUT
    :param image_bytes: an image
    :return: the prediction class (MSS/MSIMUT), the confidence of having
that prediction class
    """
    tensor = transform_image(image_bytes=image_bytes)
    out = model.forward(tensor)
    _, index = torch.max(out, 1)
    percentage = nn.functional.softmax(out, dim=1)[0] * 100
    return imagenet_class_index[index], percentage[index[0]].item()
```

Appendix I

Risk Register

Risk Register

No.	Rank	Risk	Description	Category	Triggers	Root Cause	Potential Responses	Risk Owner	Probability	Impact	Status
1	6	Losing team members	Team members leaving the team	People risk	A team member decides to leave the team	Team member's personal issue	Consult project manager, redefine task responsibilities for each remaining team member	Team	5%	High	Not triggered
2	9	Team members unable to contribute	Team member not able to complete their task responsibilities	People risk	A team member encounters some issues that affects their work	Team member's personal issue	Consult project manager	Team	10%	Medium	Not triggered
3	8	Slow decision making / Project Conflicts	Indecisive and not prioritising the success of the project	Management risk	Clash within personal interests of team members and unsuccessful understanding of given tasks.	Lack of open-mindedness and clarity within team members	Conduct internal meetings with team members and settle on a middle ground.	Team	15%	Medium	Resolved

			rather for personal gains. Unclear of project objectives and requirements.								
4	4	Delay in completion of earlier phases of project increasing failure of project completion	Not enough time to meet the schedule target to complete the project	Resource risk/ Schedule Risk	Team member does not deliver task responsibilities on time	Time estimated for a certain task is not enough	Revise schedule estimates	Team	20%	Medium	Resolved
5	1	Predictive model has a low accuracy	The predictive model developed produces wrong	Technical risk / Performance Risk	When users inputs a medical image to the predictive model	Predictive model not properly developed	Reidentify important predictors for the model, do more research on algorithms used in developing predictive model	Team	30%	High	Resolved

			outputs most of the time								
6	5	Losing source code	Source code is deleted and unable to be recovered	Technical risk	Source code accidentally deleted	Improper storage of project source code	Use GitHub for backup to minimise the risk	Team	5%	High	Not triggered
7	7	Website downtime	The platform that we decide to host our website in is unreliable and can't handle major traffics	Operational Risk	Could be caused by massive traffic or overall unreliability	Not choosing a good platform to host the website	Researching on pros and cons of several website hosting platforms and choose the platform with the least risk	Team	10%	Medium	Resolved
8	10	Slow Stakeholder Actions that delays overall project completio	Poor communication with stakeholders and lack of verbal support		Attempting to have Communicative Measures with stakeholders but stakeholders remain unresponsive	Stakeholders are difficult to get hold of	Conduct regulatory meetings and emails to stakeholders. Have a stakeholder communication plan and update it accordingly. Make sure stakeholders	Team	5%	Low	Not triggered

		n	from stakeholders				are updated through every changes in the project				
9	11	Scope Creep	Addition of unnecessary extra functionality not originally stated in the project scope	Scope Risk	Wanting to add new functionalities to enhance user experience but not addressing triple constraints of project	Excessive ideas given by team members	Clearly and succinctly state the requirements and scope in the project proposal. Update business case in case of changes	Project Manager / Team Members	5%	Low	Not triggered
10	12	Incomplete project design and deliverable definition	Not following preliminary designs created and risking to create a whole new environment that may have different deliverables	Technical Risk	Unclear preliminary designs that may not align with project scopes and definitions	Failure in understanding the given project and what is deemed an appropriate design and deliverable.	Create a preliminary design of the website, and build on it accordingly at each phase of the project.	Team	5%	Low	Resolved
11	2	Software does not	Customers are	Software Risk	Customers may not be inclined to use	When customers are not satisfied with	Ensure that a substantial amount of testing is done		15%	High	Not triggered

		fulfill customer requirements	collectively unable to obtain, view, save their results due to difficulty in accessing the user interface of the website		the website for their purposes due to it being obsolete and not user-friendly	the product	on the website				
12	3	Lack of coding capabilities leading to failure of the whole project	Lack of self-awareness in coding capabilities may result in our team abandoning the whole project	Technical risk	Lack of desire to understand the resources required to build the predictive model and website.	Lack of self-awareness on coding skills that may be exaggerated in order to	Being honest with current coding capabilities, and attending extra workshops/consultations both from Monash and online resources to ensure basic foundational knowledge.	Personal	15%	High	Resolved

10. References

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & Tan, R. S. (2017). A deep convolutional neural network model to classify heartbeats. Retrieved October, 10, 2021, from https://www.sciencedirect.com/science/article/pii/S0010482517302810

Albawi, S, & Mohammed T, A. Understanding of a Convolutional Neural network. (2017). Retrieved October, 10, 2021, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8308186

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., . . . Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, *8*(3), 292. https://doi.org/10.3390/electronics8030292

Amaya-Rodriguez, I., Civit-Masot, J., Luna-Perejon, F., Duran-Lopez, L., Rakhlin, A., . . . Brandao, P. (2021). ResNet. *Computer-Aided Analysis of Gastrointestinal Videos*, 99–114. https://doi.org/10.1007/978-3-030-64340-9_12

Ansary, U., & Sarode, T. Skin Cancer Detection Using Image Processing. Retrieved May, 12, 2021, from https://d1wqtxts1xzle7.cloudfront.net/53555135/IRJET-V4I4702-with-cover-page.pdf?Expires=1620894864&Signature=DE~Uokb0e9tF5K6Dlv8FRj5kuA4u9nf0sz5q1vqpoYikD24WSZSI8SeWbFS7DPznq6vOoLolS2OG40dqqRhOgZVPV8LMtGQqL5YdRxbf5oMoFBurAhYpW22d6YEZuql7JjdQtWlJcvfggeA4jchsWpO9kCtFuzyb-gQDVW5~6bJZxRHzGW4JvSALlBWPfm4DmrDxKp5S3iYpCqSuUWbHVNP5Z6P2qfrPVTl9VunTZ19abjwqMwTbupdvYtN5u9biJEpjW0fX0KdktcEpYwYRuEB4sLw0p9Z0xC5LOx3a4P1wF6CBVtsxaLCDiBM4msv8uw4VjLO6OhEEiC5Q4njuYw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Bhardwaj, A., Di, W., & Wei, J. (2018). *Deep learning essentials : Your hands-on guide to the fundamentals of deep learning and neural network modeling*. ProQuest Ebook Central https://ebookcentral.proquest.com

Bodavarapu, P. N. R., Srinivas, P. V. V. S., Mishra, P., Mandhala, V. N., & Kim, H. J. (2021). Optimized Deep Neural Model for Cancer Detection and Classification Over ResNet. *Smart Technologies in Data Science and Communication*, 267–280. https://doi.org/10.1007/978-981-16-1773-7_22

Brownlee, J. (2020, December 22). A Gentle Introduction to Cross-Entropy for Machine Learning. Retrieved from https://machinelearningmastery.com/cross-entropy-for-machine-learning/

Brownlee, J. (2021, January 12). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. Retrieved from https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Cancer Council Australia. (2021). Facts and figures. https://www.cancer.org.au/cancer-information/what-is-cancer/facts-and-figures

Cheng, E. S., Weber, M., Feletto, E., Smith, M. A., & Yu, X. Q. (2018). Cancer burden and control in Australia: lessons learnt and challenges remaining. Annals of Cancer Epidemiology, 2, 3. https://doi.org/10.21037/ace.2018.08.03

Delen, D. (2009). Analysis of cancer data: a data mining approach. Retrieved May 12, 2021 https://onlinelibrary.wiley.com/doi/full/10.1111/j.1468-0394.2008.00480.x

Deng, Q., He, B., Liu, X., Yue, J., Ying, H., Pan, Y., Sun, H., Chen, J., Wang, F., Gao, T., Zhang, L., & Wang, S. (2015). Prognostic value of pre-operative inflammatory response biomarkers in gastric cancer patients and the construction of a predictive model. Journal of Translational Medicine, 13(1), 66. https://doi.org/10.1186/s12967-015-0409-0

Gao, Y., & Mosalam, K.M. (2018). Deep Transfer Learning for Image-Based Structural Damage Recognition. *Computer-Aided Civil and Infrastructure Engineering, 33*(9), 748-768. https://doi.org/10.1111/mice.12363

Holcombe, J. (2020). Why should we use Flask for web development? Retrieved from https://www.quora.com/Why-should-we-use-Flask-for-web-development

Indolia, S, Goswami A, K, Mishra S, P & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. Retrieved May 24, 2021 from https://www.sciencedirect.com/science/article/pii/S1877050918308019#:~:text=Convolutional%20Neural%20Network%20(CNN)%20is,of%20traditional%20machine%20learning%20approaches.&text=This%20study%20provides%20the%20conceptual,common%20architectures%2C%20and%20learning%20algorithms.

Jafar, A., & Lee, M. (2021). High-speed hyperparameter optimization for deep ResNet models in image recognition. *Cluster Computing*. Published. https://doi.org/10.1007/s10586-021-03284-6

Jain, S. Jagtap, V. & Pise, N. Computer Aided Melanoma Skin Cancer Detection Using Image Processing. Retrieved May, 12, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050915007188

Kather, J. N., Pearson, A. T., Halama, N., Jäger, D., Krause, J., Loosen, S. H., Marx, A., Boor, P., Tacke, F., Neumann, U. P., Grabsch, H. I., Yoshikawa, T., Brenner, H., Chang-Claude, J., Hoffmeister, M., Trautwein, C., & Luedde, T. (2019). Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer. Nature Medicine, 25(7), 1054–1056. https://doi.org/10.1038/s41591-019-0462-y

Kothekar, S. (2021, September 7). What is PyTorch? Retrieved from https://www.educba.com/what-is-pytorch/

Kumar, S., Govardhan, A., & Srinivas, B. S. (2014). Data Mining Issues and Challenges in Healthcare Domain. International Journal of Engineering Research, 3(1), 857-861. https://www.ijert.org/research/data-mining-issues-and-challenges-in-healthcare-domain-IJERTV3IS10306.pdf

Liu, S., Tian, G., & Yu, X. (2019). A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter. *Neurocomputing, 338*, 191-206. https://doi.org/10.1016/j.neucom.2019.01.090

Mieczkowski, O. (2021). Flask vs Django – which one to choose and when? Retrieved from https://www.monterail.com/blog/flask-vs-django

Narang, P. (2019). When should you use cross entropy loss and why? Retrieved from https://www.quora.com/When-should-you-use-cross-entropy-loss-and-why

Narayanan, V. (2019, October 13). Tutorial — Image Classifier using Resnet50 Deep Learning model (Python Flask in Azure). Retrieved from https://medium.com/@venkinarayanan/tutorial-image-classifier-using-resnet50-deep-learning-model-python-flask-in-azure-4c2b129af6d2

Ooi, J. Liong, E., & Tan, V. (2021). *FIT3163 Project Proposal* [Unpublished assignment submitted for FIT3163]. Monash University.

Ooi, J. Liong, E., & Tan, V. (2021). *FIT3164 User Guides* [Unpublished assignment submitted for FIT3164]. Monash University.

Rezende, E. Rupprty, G. Carvalho, T. Ramos, F. Geus, P, D. (2017). Malicious Software Classification Using Transfer Learning of ResNet-50 Deep Neural Network. Retrieved October, 10, 2021, from https://ieeexplore.ieee.org/document/8260773

Shen, K. (2018, June 20). Effect of batch size on training dynamics - Mini Distill. Retrieved from https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e

Stanford University. (2021). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved from https://cs231n.github.io/transfer-learning/

11. Annex

Team Member's Contribution	
Team Members	Contributions (Sections)
Elaine Liong	1. Background - Software Main Functionalities & Software Main Features 2. Outcome - Justification, Future Works, Test Report Summary, Outcome Discussion 3. Methodology - Design & Implementation 4. Software Deliverables - What is Delivered & Usage 5. Critical Discussion - Deviation & Failure 6. Conclusion
Jack Ooi	1. Background - Literature Review & Summary 2. Outcome - Implementation, Limitation, Future Works, Test Report 3. Methodology - Implementation & Justification 4. Software Deliverables - Software Qualities 5. Conclusion
Vionnie Tan	1. Introduction 2. Background - Project Aim & Stakeholders 3. Outcome - Implementation, Results Achieved, Requirements, Justification, Discussion & Limitation 4. Software Deliverables - What is Delivered 5. Critical Discussion - Execution & Failure