Bilkent University

Department of Computer Engineering

Senior Design Project

Machine Learning for Machining Processes of Three-Dimensional Parts

Project Final Report

Project Members: Irmak Akyeli, Denizhan Kemeröz, Alp Üneri, Bulut Gözübüyük, Tuva Tanay Işıksal

Supervisor: Prof. Dr. Uğur Güdükbay

Jury Members and Project Evaluators: Erhan Dolak and Tağmaç Topal

May 6, 2022

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

The project will be an application of machine learning (ML) techniques in the field of machining process identification (MPI). ML is defined as a branch of artificial intelligence (AI) and computer science that focuses on the use of data sets and algorithms to mimic the way humans are able to learn on computer systems [1]. ML algorithms, often called models, are fed data from data sets that gradually get better and better in their accuracy in the same ways that humans would [1]. For example, an ML model may be constructed to identify how many faces are in a picture that is supplied. In order to do this, the model would have to be trained using a known data set, which involves feeding the model data on which the aspect in question is known, and the model would then gradually get better and better at identifying how many faces there are in a supplied picture. After sufficient training, the model would be able to identify the number of faces within a picture with great confidence.

MPI using ML is a novel area of research that aims to automate the manufacturing of mechanical parts by automatically deciding whether the part in question would be able to be manufactured using the manufacturing plants in place in an effort to make the manufacturing process more time and cost-effective and to reduce the number of faults that may occur during manufacturing [2]. Our project will investigate the applicability of ML and especially convolutional neural networks (CNNs) for MPI means. A CNN is a deep learning model that takes in an input image and by assigning importance such as learnable weights and biases to various aspects or objects within the image is able to differentiate images within a number of categories the aspects/objects within the image can belong to [3].

We will be trying to come up with an ML model that will be used to automate the processes of determining the type of manufacturing processes (additive versus subtractive manufacturing) to be used, the producibility of three-dimensional models, and cost estimation. Additive manufacturing is when processes build objects by adding materials layer by layer to form the desired product, while subtractive manufacturing is when materials are removed from an already existing object to come up with the part that is needed [4]. We are planning on developing a deep learning framework for determining which type of machining processes are suitable for producing a machine part and the producibility of these parts using the selected machining process for the provided three-dimensional models.

The machining operations we will consider are turning, milling, and drilling. Turning is a machining operation in which a workpiece is rotated while the cutting piece moves in a linear motion to achieve the desired shape, which is often cylindrical [5]. A lathe machine is usually used for turning purposes [5]. Drilling is a machining operation in which a drill press or a tapping machine is used to create a round hole in a workpiece [5]. Lastly, milling is a machining operation that involves using many multi-point rotary cutters to remove material from an object to achieve the desired outcome [5]. Our end product will also be able to determine which operations are to be performed in order to attain the desired part if the part is able to be produced using the available producing plant. Another stage for the parts that are producible using the selected machining process and the operations selected is the estimation of the cost of the production of the parts for different material properties and costs input to the system.

2 Requirements Details

In this section of the report we will be discussing the requirements of the project in detail. We will first start with the functional requirements and move on to discuss the non-functional requirements.

2.1 Functional Requirements

In this section of the report we will be discussing the functional requirements of the project in detail. We will first be focusing on the functionalities of the model that will be trained before it reaches the user, and will then be delving into the functionalities that the end user will have access to.

2.1.1 Functionalities of the ML Model

The model would be able to be trained by supplying it with data in the mode of binvox files. The model will have to first be sufficiently trained in order to then be able to predict and perform operations on user supplied parts.

This training process requires long computations and depending on the CPU used can have a runtime of as little as two seconds for each epoch on an NVIDIA RTX 30 series with a total of 15 hours to completely train the model. However, it should be noted that this training will be done only once and before the project reaches its end user. Thus, the length of the duration of this process will not be of issue to the end user.

2.1.2 Functionalities of the User

The main functionality of our project from an end user perspective is the trained model being able to predict the user supplied part models and accurately determine whether it is able to be produced using the given manufacturing plant, what materials will be necessary to produce the part, the method of manufacturing and the machining operations that will be required to manufacture the part, and the estimated cost that will be required to produce this part. In order to do this, the user will be able to supply to our program an STL file, have it transformed into a binvox file, have this transformed into a numpy array, fed into the model, and have the respective output be produced. The user will be able to supply models in the form of STL files and will be able to view this model turned into a binvox file and the output resulting from the given input. The model will then predict whether a part of the given model can be manufactured using the given manufacturing plant or not, what method and manufacturing and machining operations will be used to manufacture the part, and the estimated cost of doing so.

2.2 Non-Functional Requirements

In this section of the report we will be discussing the non-functional requirements of the project in detail.

2.2.1 Usability

As our end-product will only have to be used by professionals within the field of manufacturing, we will not be required to cater to a general audience when creating our project. Thus, we can assume that the people using our project will be familiar with the technical terms within our project such as turning, drilling, and milling, and we will not be required to provide a description of such terms. We will also be able to assume some level of expertise on the abilities of the people using our project to be able to interact with computer systems, as we are expecting the large majority of our users to be engineers. As such, we will be able to assume the users will be familiar with using systems such as the one we will develop, and can be more relaxed in the way we approach the user interface elements et cetera.

2.2.2 Supportability

Our project will be able to be run on any system that runs Python3 excluding 3.8+ as it does not support the Tensorflow library that we are using for our CNN models, and we will be focusing on the interoperability of the underlying Python fundamentals to establish the supportability of our project.

2.2.3 Reliability

We aim for our ML models to be able to be accurate to a very high degree after sufficient training. We at the moment are unable to estimate the size of the data set that will be required to achieve this, but we will be putting the bar at around 95% to be able to call the model accurate.

2.2.4 Efficiency

As the model will have to be trained only once for it to be able to output accurate classifications, efficiency will not be the biggest non-functional requirement in our project. We will be putting more emphasis on our project producing more accurate results and will be ruling in favour of losing efficiency for this aim if need be.

2.2.5 Security

As there is no sensitive information that is to be stored or processed in our project, there are no security concerns for data leak or any such considerations. Therefore, no encryption or any other method of obfuscation will be necessary in our project.

2.2.6 Scalability

Our project will be taking into consideration the machining operations of turning, drilling, and milling. Thus, it will have to be scaled if need be to have any other machining operations that are to be incorporated. We cannot comment at this time how easy or difficult it will be as we do not yet have the model, but we are assuming it will be easier to implement such a scaling to our project.

3 Final Architecture and Design Details

We will be sticking to a modified version of the Model-View-Controller design pattern while decomposing our system into subsystems. Our project will not require any additional hardware to run. We do not have any persistent data that has to be maintained throughout our project. Our project does not have any security risks associated with it. Our project will work in a strictly sequential manner, therefore we do not have any global software control. We will have three boundary conditions; initialization, termination, and exception handling.

3.1 System Models

In this section of the report, we will be discussing the use cases and the scenarios of our system, and will as well be providing UML diagrams for the system.

3.1.1 Scenarios

Scenario	Uploading the STL File
Participating Actor	User, system
Flow of Events	The user selects an STL file to upload to the system. The selected STL file is uploaded to the system.
Entry Condition	The user presses the “Upload STL File” button.
Exit Condition	The user presses the “Upload” button.
Quality Requirements	The system must not allow the user to upload any other type of file. The system must accurately save the STL file.

Scenario	Selecting the Material
Participating Actor	User, system
Flow of Events	The user views the available types of material within the system. The user selects a material out of the displayed list.
Entry Condition	The user presses the “Select Material” button.
Exit Condition	The user presses the “Proceed” button.
Quality Requirements	There must be a suitable number of materials that the user can choose from.

Scenario	Convert the STL file into BinVox and Display the Model
Participating Actor	User, system
Flow of Events	The supplied STL file is converted into BinVox. The BinVox model is displayed to the user. The user checks the converted BinVox model.

Entry Condition	An STL file must be uploaded and a material must be chosen. The user presses the “Convert to BinVox and View Model” button.
Exit Condition	The user either presses the “Confirm” or the “Go Back” button.
Quality Requirements	The BinVox model must be displayed clearly. The BinVox model must be able to be viewed in a three-dimensional manner with the user being able to rotate the model et cetera.

Scenario	Predict the Model and Display the Results
Participating Actor	User, system, display
Flow of Events	The user asks for the system to predict the BinVox model and display the results. The system does so.
Entry Condition	A BinVox file that has been approved by the user must be present within the system. The user presses the “Predict Model and Display the Results” button.
Exit Condition	The user presses the “Close” button.
Quality Requirements	The system must be able to predict the model with sufficient accuracy (above 95%).

3.1.2 Use Case Model

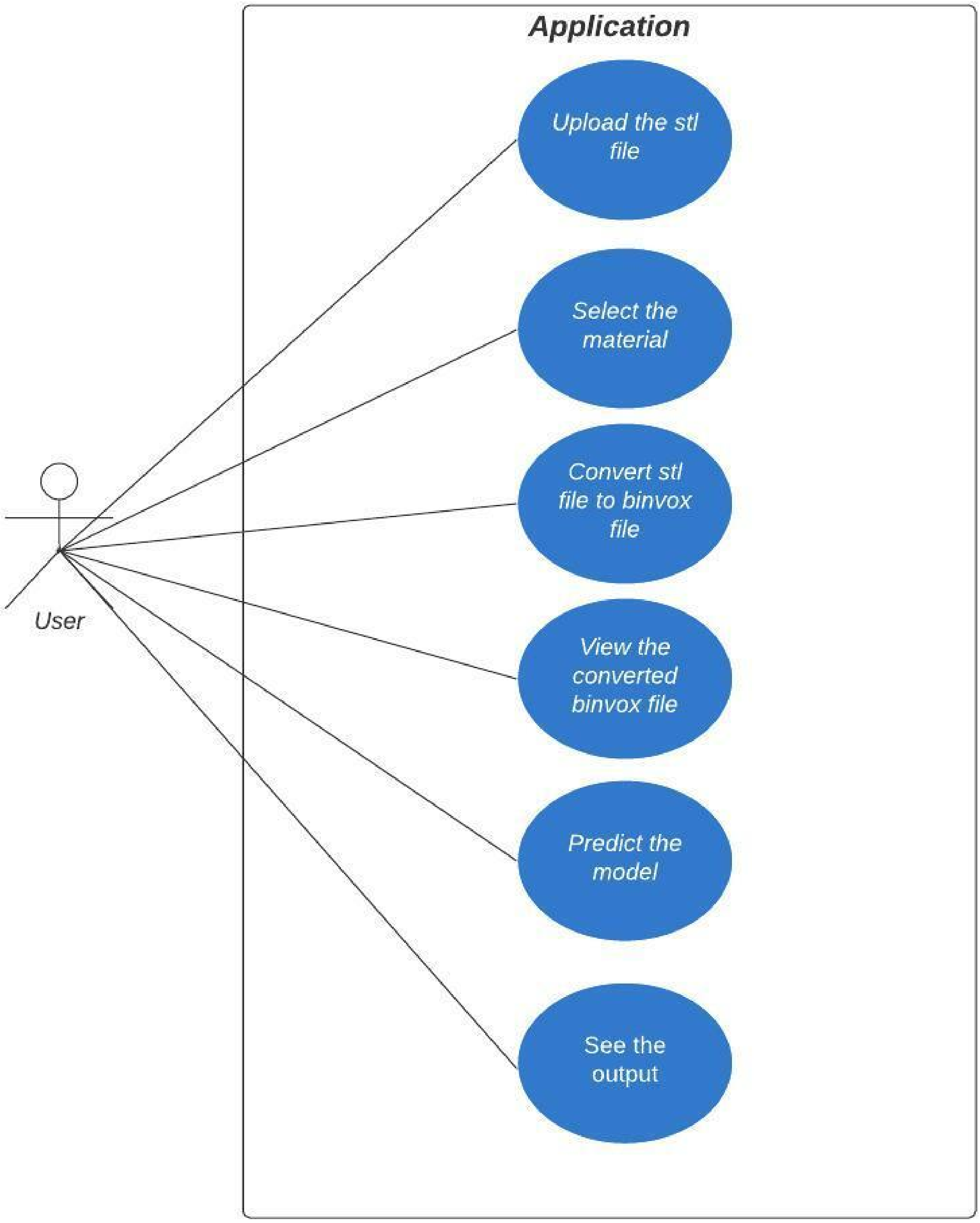


Figure 1: Use Case Diagram

The use cases within the model are explained in detail under the “Scenarios” section.

3.1.3 Object and Class Model

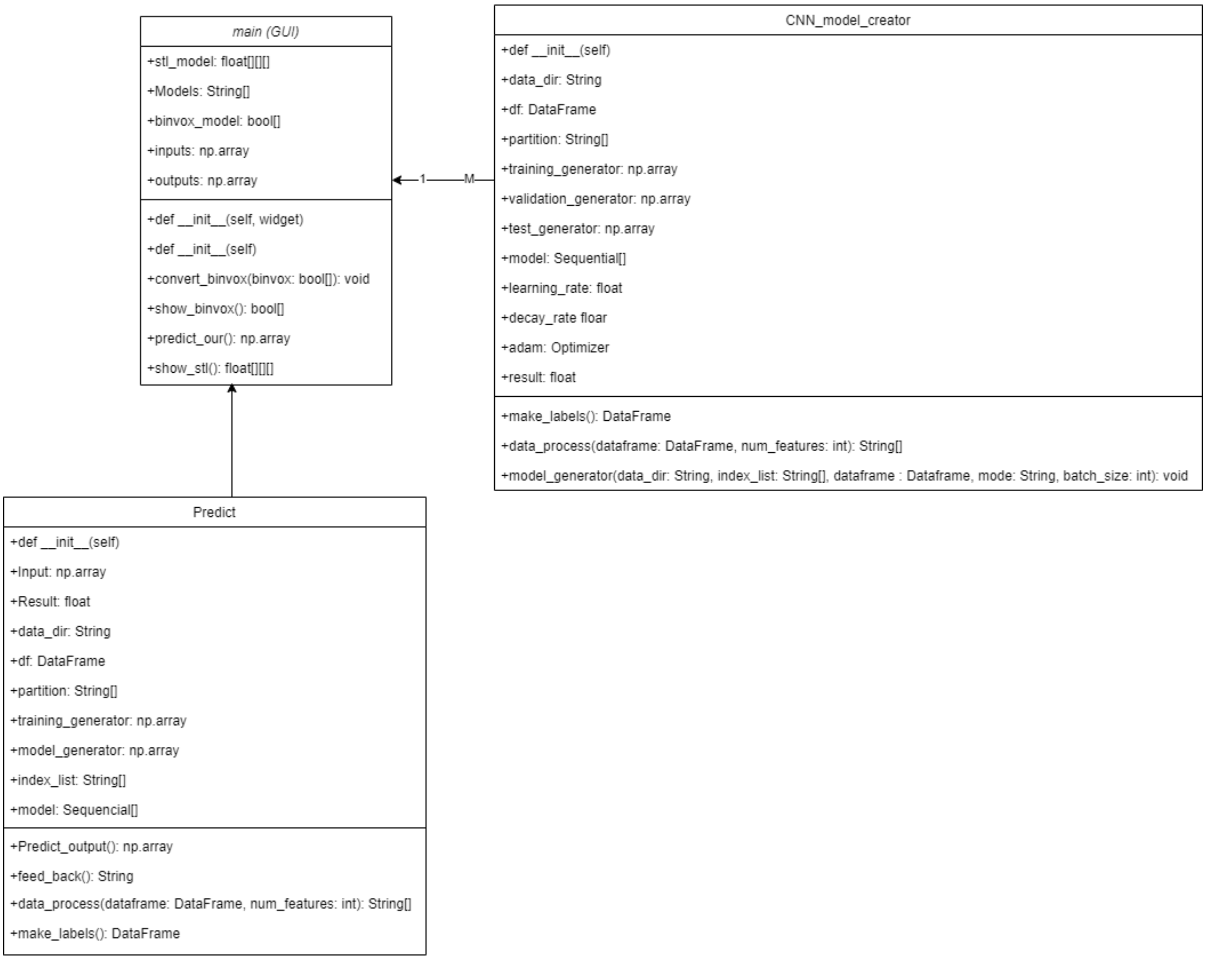


Figure 2: Object Class Model Diagram

3.1.4 Dynamic Models

In this section, we will be discussing the state machine diagram, the sequence diagram, and the activity diagram of our system.

3.2.4.1 State Machine Diagram

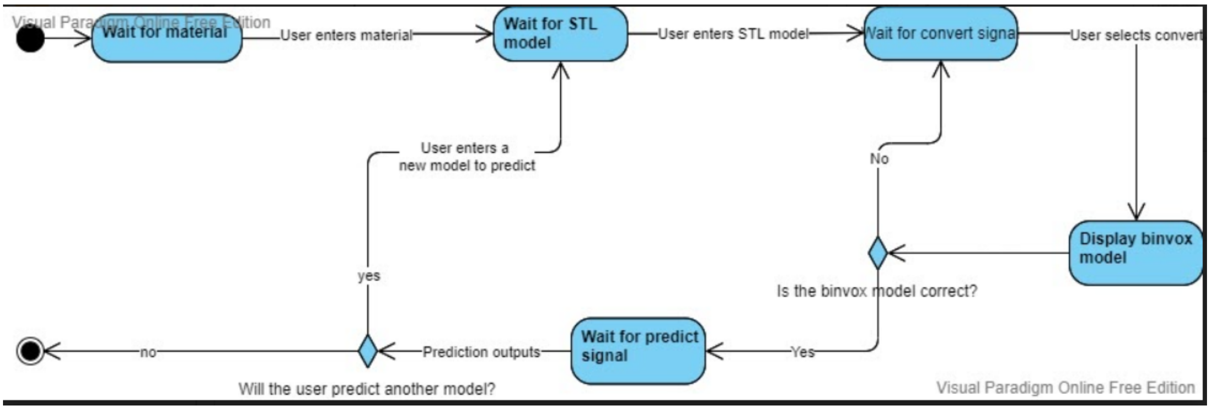


Figure 3: State Machine Diagram

The application starts with the user selecting a material from the list of materials that are available within the system. The user then uploads a three-dimensional model of the part that is to be manufactured in the format of an STL file. Next, the system converts this file into BinVox format and displays this file for the user to check. The user then either confirms the BinVox file or goes back to reconvert the file into BinVox. Next, the user clicks the predict button and the system predicts and displays the results. If the user wishes to predict another model, they press the button to go back to the main menu and go through the explained process once more.

3.1.4.2 Sequence Diagram

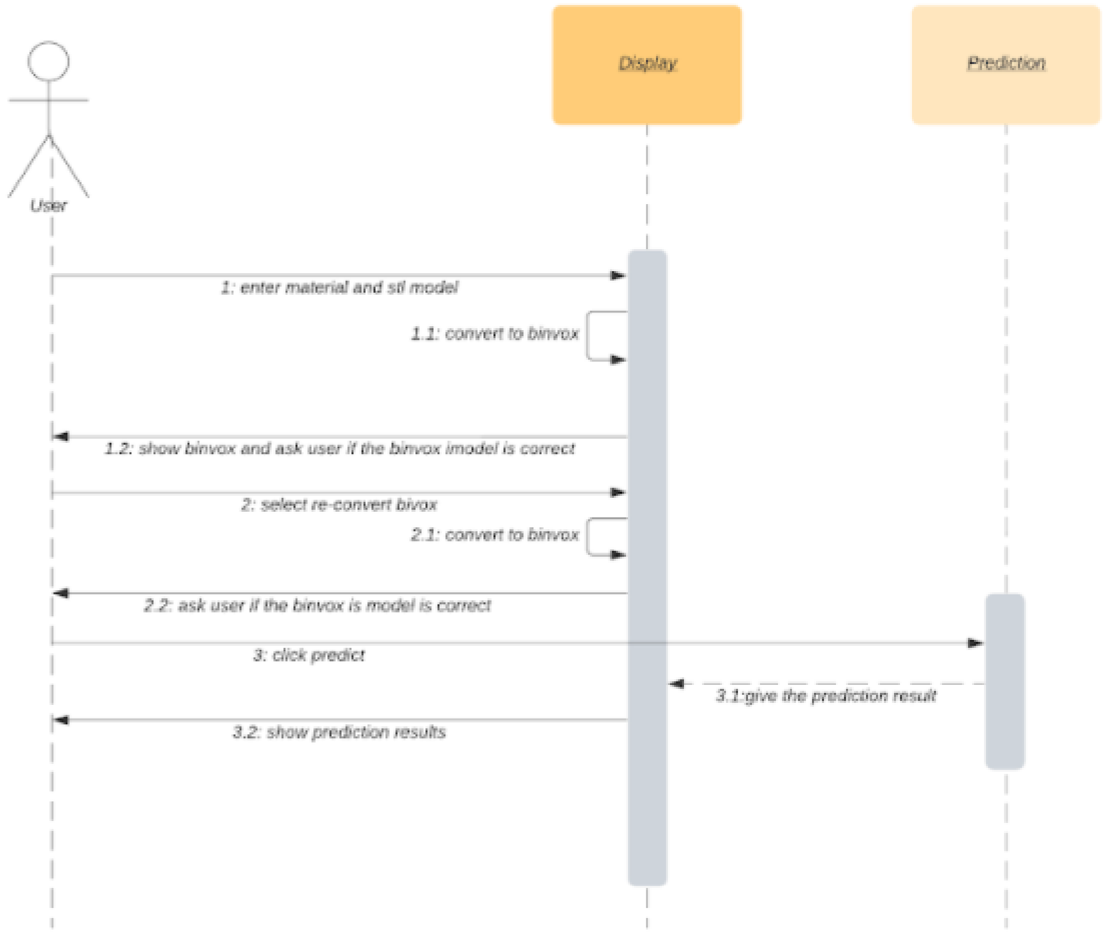


Figure 4: Sequence Diagram

The user first chooses the material that the part will be made out of from a list of materials available within the system and uploads an STL model to the system. This is done through the GUI. This STL file is then converted into a BinVox model and this model is shown to the user. If the user spots any problems within the model and wishes to reconvert it into BinVox he does so using the GUI. After the user confirms the model, they click the “Predict and Display Results” button via the GUI and the system predicts the model using the machine learning backend of the system. The machine learning model then passes the results to the display, and they are shown to the user over the GUI.

3.1.2.3 Activity Diagram

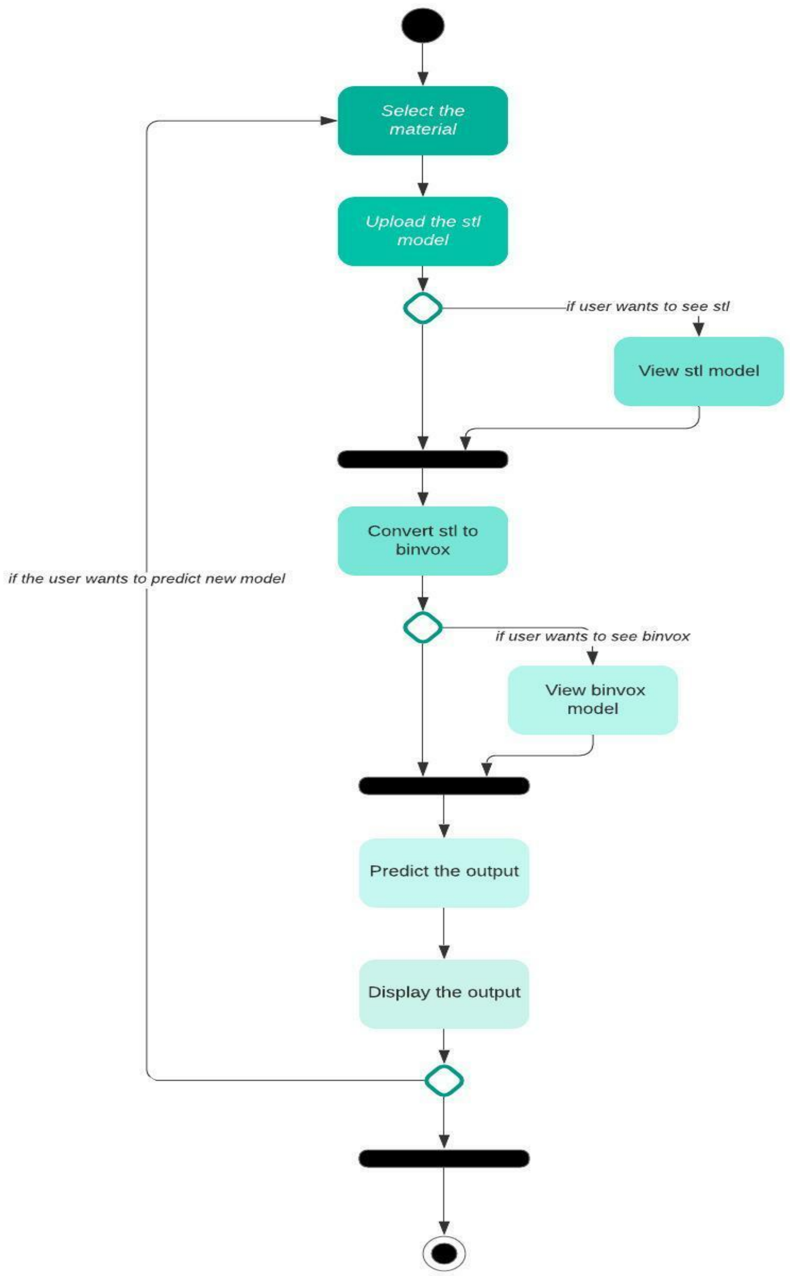


Figure 5: Activity Diagram

The user will first select the material that the part will be made out of out of a list of materials that are available within the system. Next, the user will upload an STL file to the system and will be able to view it if they wish to do so. The user will next convert this STL file into BinVox format, and will likewise be able to view the BinVox model should they want to. Next, the system will predict the model and display the output. If the user wishes to predict another model, the process will start once more.

3.1.5 User Interface

Our UI will have four main components, on the top left we will have the material selection and STL file upload sections, on the bottom left we will have the buttons for the use-cases of our program, on the top right we will have the prediction results, and on the bottom right we will have our model viewers.

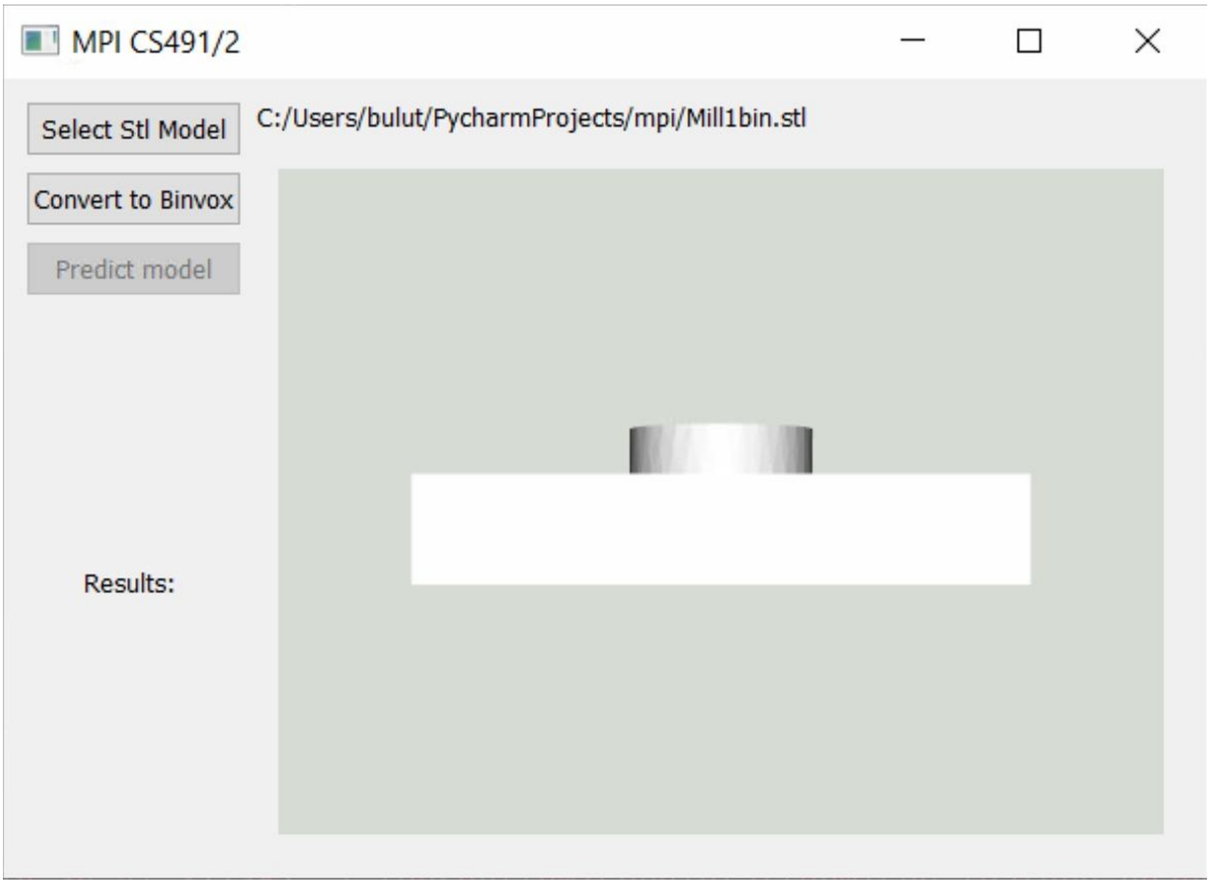


Figure 6: Interface After Selecting STL File

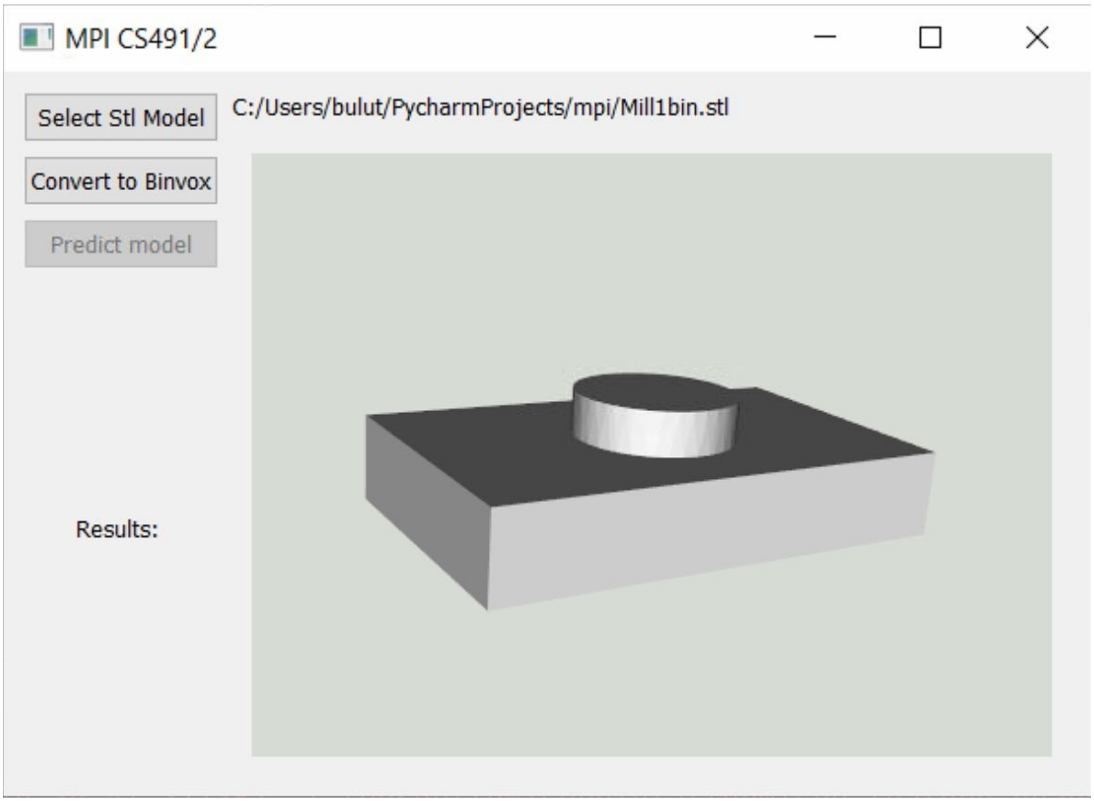


Figure 7: Interface Available to See STL Model from Every View

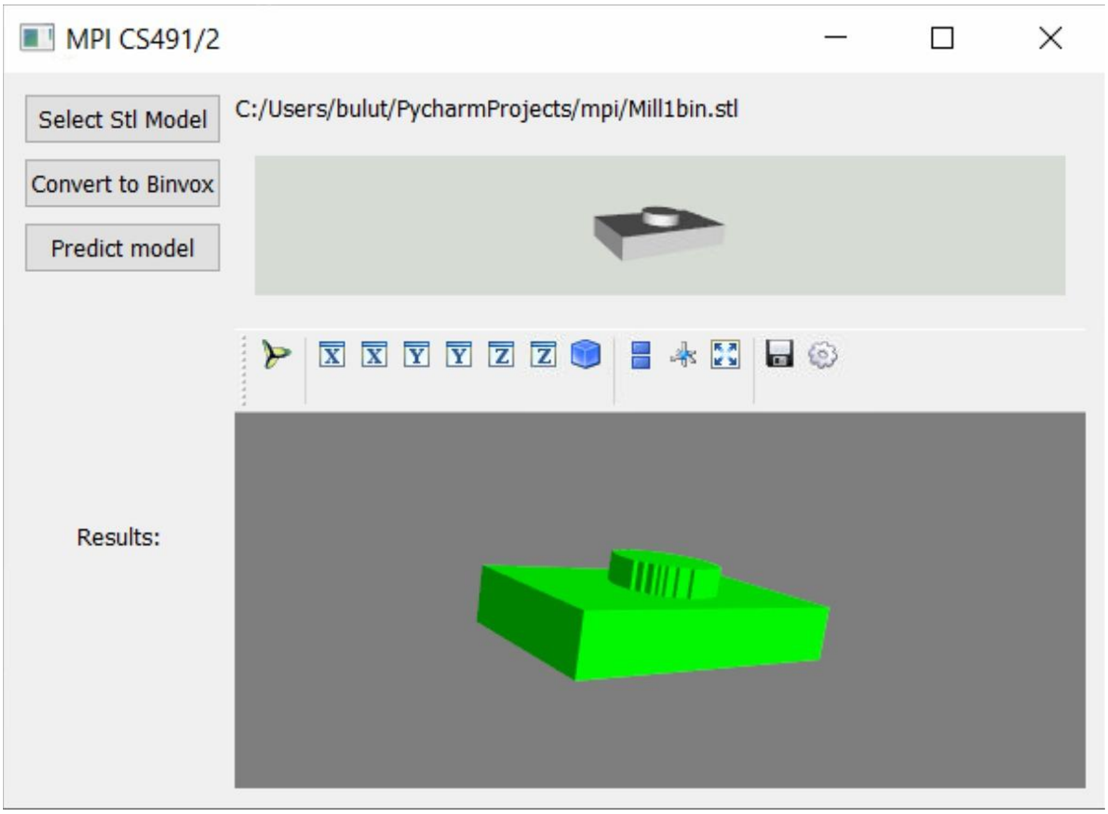


Figure 8: Interface After Converting the STL Model to Binvox

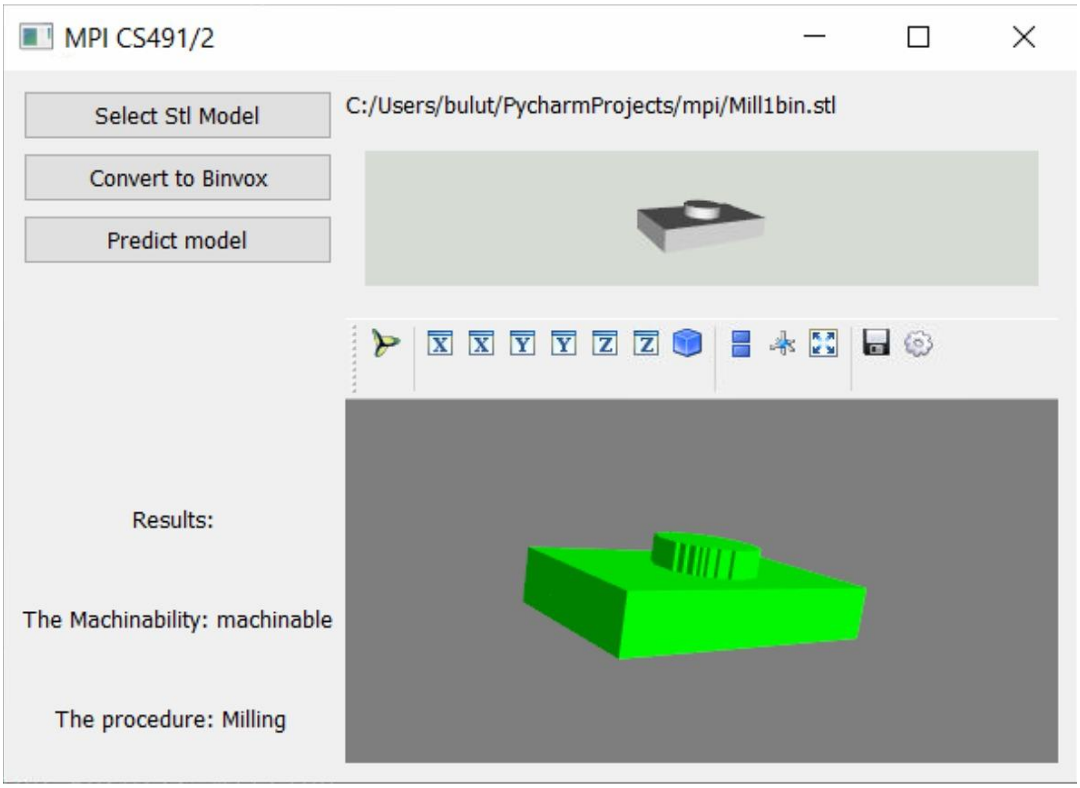


Figure 9: Interface After Predicting the Results

3.2 Subsystem Decomposition

Our system will be divided into three subsystems, called the UI Management Subsystem, Core Logic Subsystem, and the File Management Subsystem. In our subsystem decomposition we have decided to follow a design approach inspired by both the Model-View-Controller design pattern and the layered approach, however we have not stuck to either of them fully. We have essentially decomposed the system into a subsystem that implements both the Model and the Controller in the MVC approach as one subsystem (our Core Logic Subsystem) and a separate View subsystem (our UI Management Subsystem). We have as well a third subsystem that will be handling the uploading of STL files onto our system and the saving of the BinVox files onto the hard drive of the user, called the File Management Subsystem.

The services associated with these subsystems will be elaborated further on Section 4. However, the following is a quick summary rundown of the main functionalities of the three subsystems.

3.2.1 UI Management Subsystem

This is the subsystem that corresponds to the View subsystem in the MVC design pattern. The main service this subsystem will provide is creating and managing the GUI of our project. It will have different components such as STL and Binvox model viewers and a prediction output viewer screen.

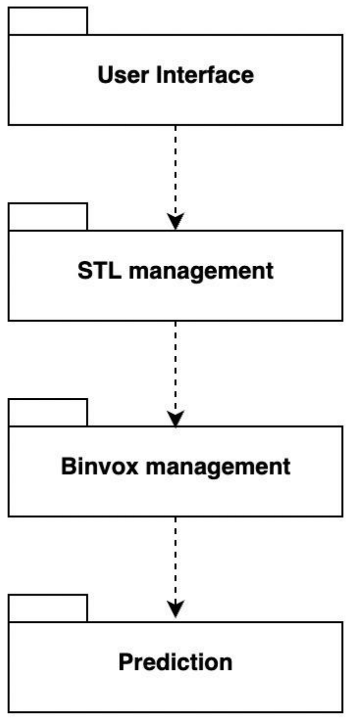


Figure 10: UI Subsystem

3.2.2 Core Logic Subsystem

This is the subsystem that corresponds to the Model and the Controller subsystems in the MVC design approach. As our project is of a relatively compact scale, we have decided not to separate these into two subsystems and establish a communication between them but have rather decided to do it all in one subsystem. This subsystem will mainly be responsible for two use-cases; converting the supplied STL model into Binvox format and predicting the Binvox model.

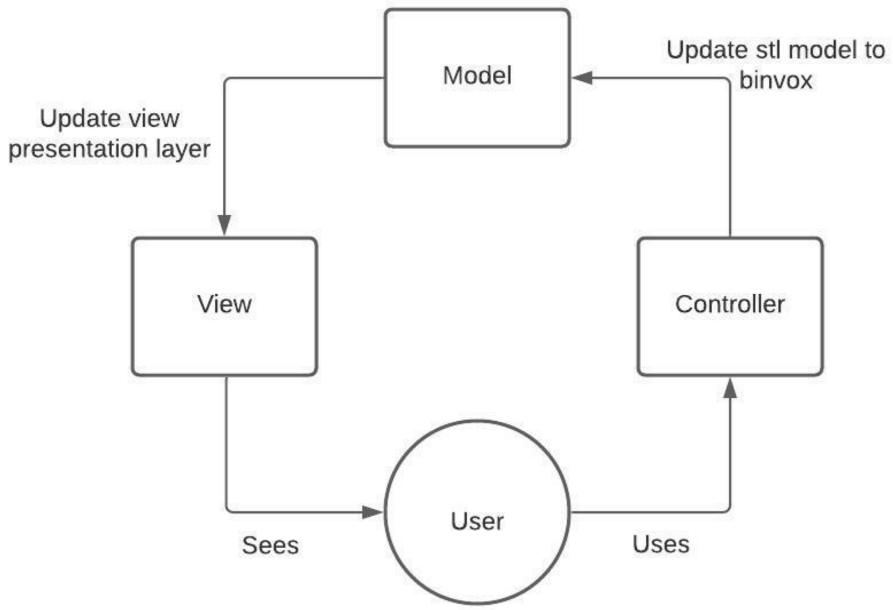


Figure 11: MVC paradigm

3.2.3 File Management Subsystem

This is the subsystem that will be responsible for the uploading of the STL files onto our system by the user and the saving of the Binvox files that have been created by conversion from the STL files onto the hard drive of the user. We have decided to have this as a separate subsystem as we believe these functionalities do not really fall under any of the categories in the MVC design model. This will be a fairly basic and simple subsystem.

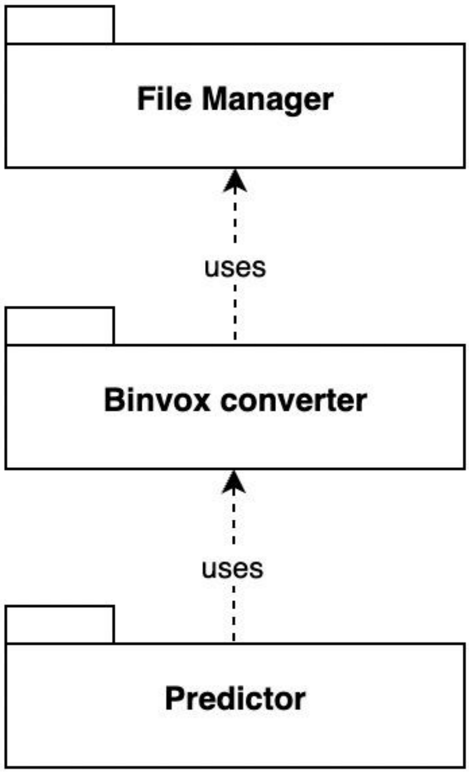


Figure 12: File Management Subsystem

3.3 Hardware/Software Mapping

We will be implementing our project using the Python programming language. Our project will not require any additional hardware on top of the standard keyboard/mouse set up to be run. For the most part, users will only have to interact with our project using the mouse.

Our ML model will have to be trained before our product is ready to be used, but this will only have to be done once before it is ready. In order to speed up this process of training our model, a computer with high specs would be preferable. However, since this training will only have to be done once and out of the sight of the users, it is possible to train our model on virtually any machine.

We will be using the hard disk space of the user to store our files such as our trained ML model. We will not require any installation process before the user is able to run our project.

3.4 Persistent Data Management

Our project will not require any complex data storage system or database in order to be used. We will be dealing with a couple types of files as part of our project, namely STL and Binvox files. The users will have the option to upload any STL files and have them converted to Binvox as well as view these Binvox files. These files will be stored on the hard drive of the user and our system will not be handling any persistent data when it comes to these files. If a user wishes to save one of their files onto their hard drive, they will be able to do so, yet our project will not check for any persistent data when doing so.

3.5 Access Control and Security

There will be no networking component to our project. As such, we will have a minimal attack surface to our program. We will not be storing any sensitive data on our system either, so sensitive data leakage will not be an issue. We will not have any login or authentication system to our project, as we do not require one. Our project will essentially be an STL to Binvox converter and a predictor of several attributes on a Binvox file. It does not have any surface for any meaningful attacks.

3.6 Global Software Control

Our project will be run in a strictly sequential fashion, as shown in our state machine diagram in our analysis report. Since we do not have any concurrent events, our global software control will be easy. We will be moving in sequential order through our use-cases. Our system will move from one use-case to another via the user clicking a button. Therefore our system will be event-driven. We are not going to be using separate threads for each event, our program will run in its entirety as a single thread and by extension process.

3.7 Boundary Conditions

In this section of the report, we will be discussing the boundary conditions of our project. We have three boundary conditions; initialization, termination, and exception handling.

3.7.1 Initialization

Our project will not require any installation process to be usable. We will only require that the user has the appropriate Python version and packages. The system will be created whenever our executable is executed. Upon launching our program, the user will be met with the main screen. On this screen they will be able to select a material to work with and will as well be able to upload an STL model. From this point onwards they will have the ability to view the given STL model, convert the model to Binvox, view the Binvox model, and predict the Binvox model.

3.7.2 Termination

The user will be able to terminate the program at any time by clicking the exit button. Once they do so, the system will terminate. During this termination process, the system will check if there are any ongoing predictions, and will ask the user if they wish to terminate mid-prediction or not. If they wish to do so the system will cancel the prediction and quit. If not the system will abort the termination and move on as usual.

3.7.3 Exception Handling

If the user tries to upload a file that is not in STL format when trying to upload an STL model, the system will reject the file. Further, if there are any exceptions that present themselves during the conversion of an STL file into Binvox format, during any predictions or while attempting to view a model, the system will display an appropriate error message and terminate. Since we do not have any persistent data that is to be stored on our project, it is okay for our system to simply terminate after displaying an appropriate error message.

3.8 Subsystem Services

In this section, we will be discussing in more detail the services of the subsystems that we have outlined under Section 3.2 of our report. Here, we will delve into further discussion regarding the services of our subsystems.

3.8.1 UI Management Subsystem

As previously discussed, this subsystem will be tasked with handling the creation and the management of the GUI of our project.

3.8.1.1 Viewing STL and Binvox Models

This will be the main functionality of this subsystem. One of our core use-cases is allowing the user to view STL and Binvox models. This subsystem will be responsible for the displaying of such models. It will display these models on a separate window in a three dimensional manner with the user being able to rotate the model in any direction they please as well as zoom in and out to the model.

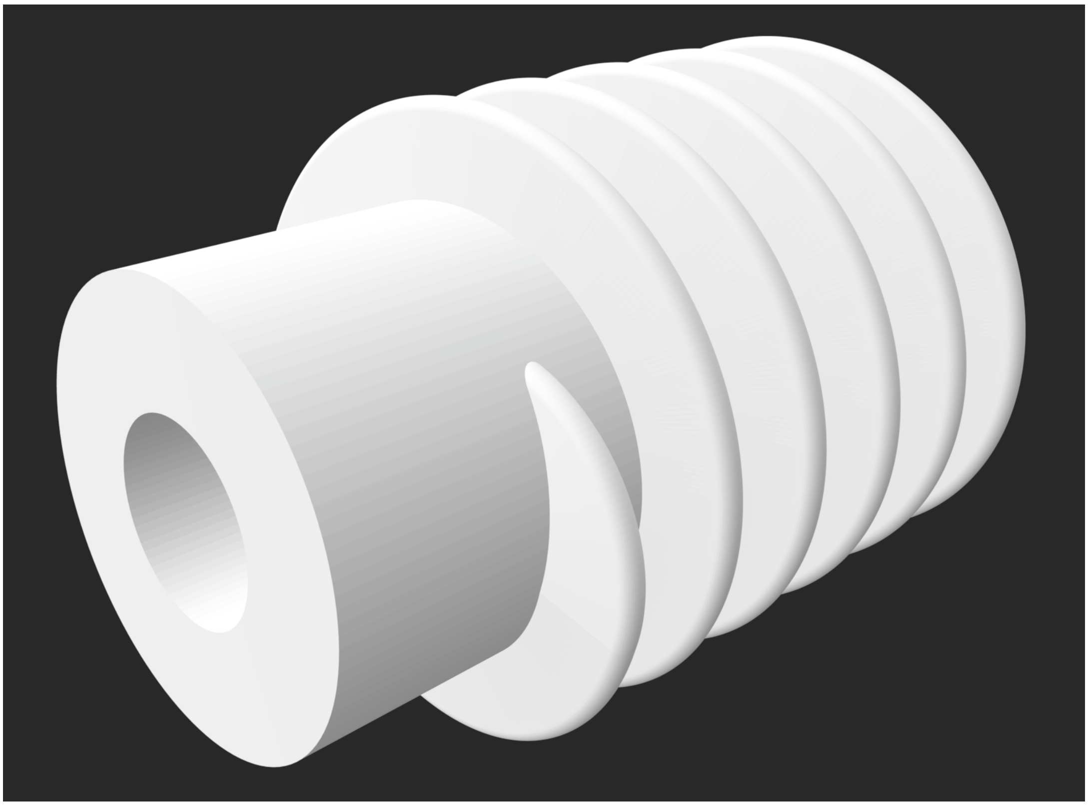


Figure 13: Example of a stl model

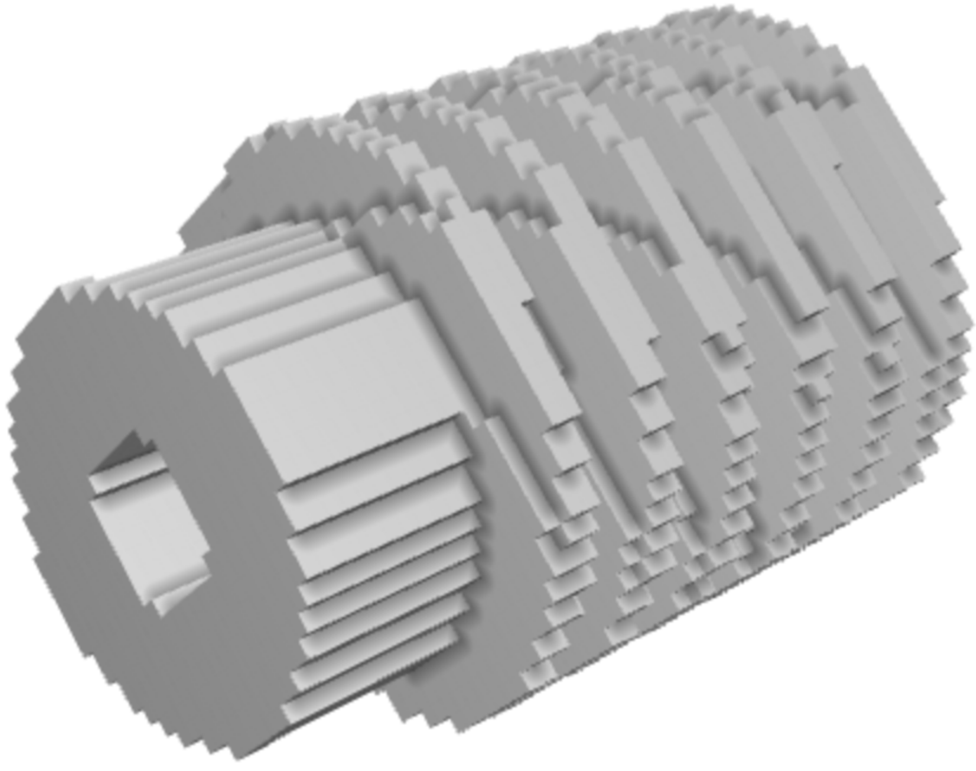


Figure 14: Example of a binvox

3.8.1.2 Viewing Core Screens of the System

This will be the secondary functionality of this subsystem. As the subsystem corresponding to the View subsystem in the MVC design approach, this subsystem will be responsible for the creation and the displaying of all the screens within our project.

3.8.2 Core Logic Subsystem

This is the subsystem of our project that corresponds to the Model and Controller subsystems within the MVC design paradigm. As previously stated, since our project is of a relatively compact size we have decided not to split this into two separate subsystems in favor of more rapid communication and deployment.

3.8.2.1 Converting STL Files into Binvox Format

This will be the secondary feature of this subsystem. It is a prerequisite that has to be performed before the main feature of the subsystem and as well the main feature of our project is able to be performed. This subsystem will be tasked with taking the STL file supplied by the user and successfully converting it into Binvox format. The user will then be able to assess the correctness of the Binvox model using the UI Management Subsystem and proceed to the main feature of our program if they are satisfied with the conversion.

3.8.2.2 Predicting Binvox Models

This will be the main feature of this subsystem and as well our project. This subsystem will be tasked with implementing the prediction feature of our system, which is our main feature. Given a Binvox model of a part that is to be manufactured and a manufacturing plant, our project will be able to predict whether the given part can be manufactured, with which machining operations it can be manufactured if it can be manufactured, and the estimated cost of doing so. This subsystem will be using the trained ML models of our project in order to predict these and display the results.

3.8.2.3 Training CNN Models

The training of the 3D CNN models consists of adding different layers of Keras Library Sequential model layers, manipulating parameters and optimizing the result. In our case we used a few different model layering for our different prediction models, in which all layers serve different purposes to obtain the best result possible. These systems are highly dependent on the training data and require a great number of them to train with. For the moment our training depends on the data that we took from [2] and we plan to later on add more taken from different sources to differentiate the functionalities of our project. The training of models is done once as mentioned previously and this part of the code will not be included in the source code provided to future clients.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d (Conv3D)              (None, 29, 29, 29, 32)    11008
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 29, 29, 29, 32)    128032
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 29, 29, 29, 64)    55360
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 15, 15, 15, 64)    0
_________________________________________________________________
flatten (Flatten)            (None, 216000)            0
_________________________________________________________________
dropout (Dropout)            (None, 216000)            0
_________________________________________________________________
dense (Dense)                (None, 128)               27648128
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 16)                2064
_________________________________________________________________
dense_2 (Dense)              (None, 21)                357
=================================================================
Total params: 27,844,949
Trainable params: 27,844,949
Non-trainable params: 0
```

Figure 15: Example layers of the first CNN model, feature classification

3.8.3 File Management Subsystem

This subsystem will be tasked with managing the upload and save features of our project. It will be tasked with allowing the user to upload STL files onto the system and saving Binvox files onto the hard drive of the user.

3.8.3.1 Uploading STL Files onto the System

The user will be able to upload an STL file containing the three dimensional model of the part they wish to predict onto the system. This subsystem will be in charge of this functionality.

3.8.3.2 Saving Binvox Files onto the Hard Drive of the User

The user will as well be able to convert these STL files it uploads onto the system into Binvox format and save them onto their hard drive. This subsystem will be in charge of saving the Binvox files onto the hard drive of the user.

4 Development/Implementation Details

As we have explained before, our project is not an object oriented project. Therefore we do not use components like interfaces and abstract classes. Instead, we have a main class, a prediction class and a training class. From the nature of our project, the main class accesses the predict class when the user clicks the predict button. The training class is not linked to any other classes and is not accessed by them. Its sole purpose is to create trained CNN models that we will provide to the customer and the class itself will not be provided in the source code. Having only 3 classes the implementation packages were not necessary.

In our project however, we have a 3D CNN model training system that uses many different libraries provided by Python that should be mentioned to understand the training process. A brief summary of the libraries used and their descriptions are as follows:

- Numpy: Numpy is a powerful python library that enables the user to work and operate on multidimensional arrays. In our project numpy is used to store binvox files transformed to multidimensional arrays and is the input type of our CNN training models.
- Tensorflow: Tensorflow is an open-source python library for ML and AI. It is an easy to use tool that lets the user create many different ML models such as deep-learning models like CNN. Keras is also a specialized version of tensorflow that focuses on deep-learning.
- Keras: Keras is an interface implementation for tensorflow that simplifies the use of the library in order to achieve better results and understanding. In our project the implementation of our CNN models are made with tensorflow, keras. From Keras we are using layers: Input, MaxPooling3D, MaxPooling2D, Dense, Flatten, BatchNormalization, Dropout and Conv3D. As well as its optimizers and regularizers.
- Pandas: Pandas is a python library created for data manipulation and analysis. In our project pandas is used to create training and validation partitions of our data.
- Binvox_rw: Is an open source code made for python that enables the user to work on binvox files and manipulate them to create different variable types. In our project as the name suggests binvox_rw is used to manipulate our binvox files and turn them to numpy arrays.

Like mentioned in the previous reports about our project, our aim is to improve the accuracy of our source code, add aspects missing in it and turn it to a working

MPI system. To do this we need to analyze and understand the training model provided. We currently have three 3D CNN networks for three different tasks. The first one takes 21 different feature used in machining and categorize them under turning, milling and non-machinable features, the second one takes a machinable data input and predicts its machining procedure that can be milling, turning or both of them and finally the third model takes a data input and decides if it is machinable or not. The first CNN network, as you can see in Figure 15, operates on 200 distinct part models that are each taken from 6 different orientations for each feature. This makes a total of 20.000 models that are used in the creation of the CNN network. 70% of this data is used in the training, 15% of it in the validation and the remaining 15% is used in the testing. The CNN model is created using keras’s sequential model, a model used for layering with single input and output of each layer, and has 10 layers. As the results of the other two models are dependent on the features provided, this CNN network is the most detailed one and is trained on 20,000 epochs. The Second and third CNN network, as you can see in Figure 16, is initialized with the weights loaded from the first network (the weights obtained after the training on 20,000 epochs) so that they are not required to train from the beginning. The second CNN network is fed 500 milling, 500 turning and 500 milling-turning models again all taken from 6 orientations. With the weights already loaded from the first network, only fine tuning is performed during the training and 500 epochs is used to achieve this. The last network takes the input data of the second network as machinable models and takes another 2400 models as non-machinable models. Again similar to the second network the weights are loaded from the first network and only 500 epochs are used to train the network. All the three networks show over 95% accuracy on the training.

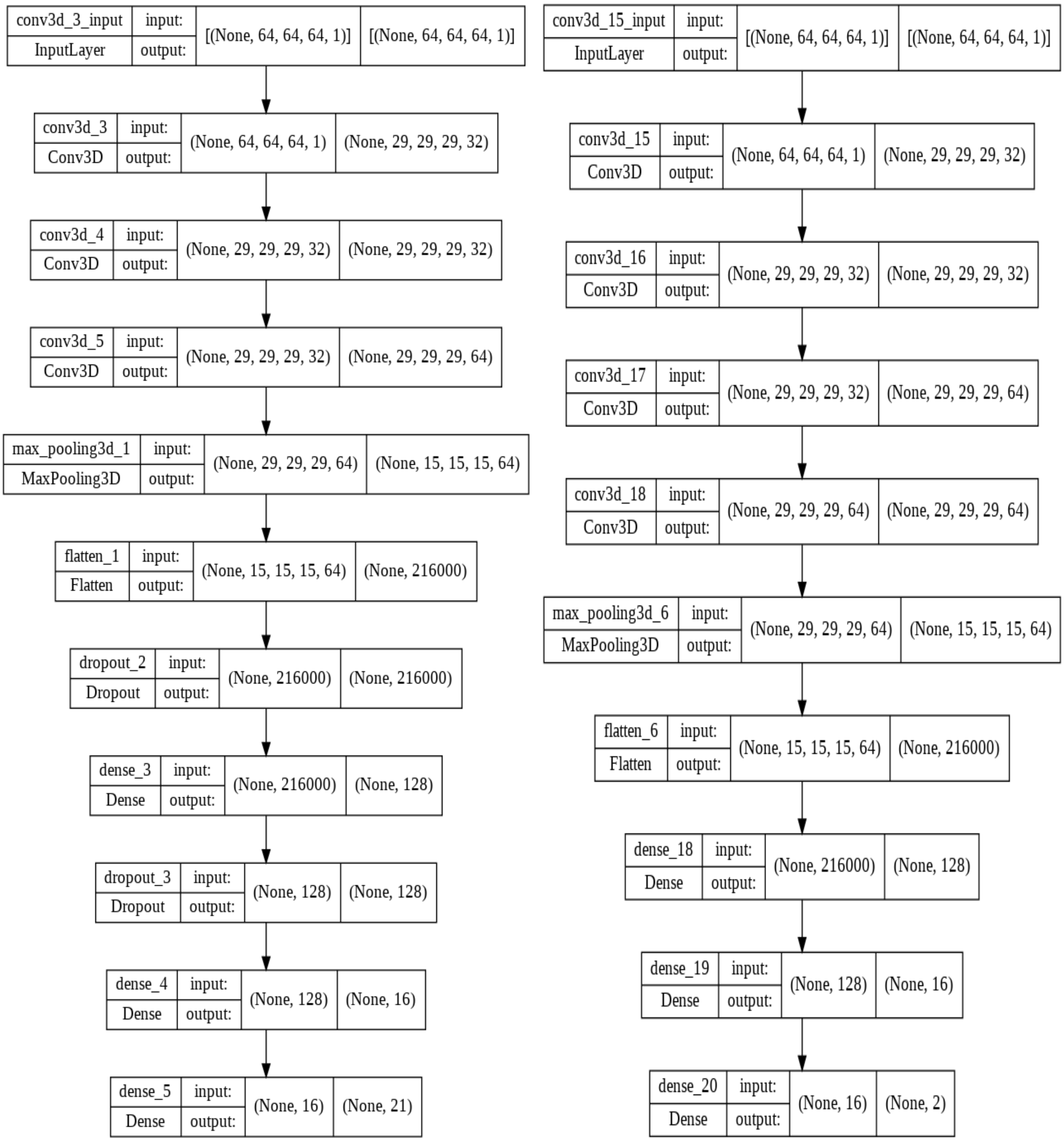


Figure 16: Example Layers of the Second and Third CNN Network

5 Testing Details

Contrary to the accuracy shown during the training outputs, when the model is tested on the real parts, concerning results are obtained and we are currently trying to improve them. The network shows a tendency to predict milling-turning for the machining parts that should be only turning and almost never predicts turning alone. Likewise, while looking at the machinability three out of 8 models returns wrong predictions and when the same binvox file is used to predict multiple times, the result also varies. All these makes the networks unreliable and in need of improvement. There are several possible reasons as of why the user receives these predictions:

- The data used in the training may not be enough for small enough details or accurate predictions.
- The resolution of binvox files that is currently 64x64x64 for competitive purposes is not enough for distinction of different shapes that we are trying to recognize as features, and tempers with the smooth corners too much to make them inseparable from sharp corners. Differences can be seen in figures 13 and 14.
- The different numbers of features and models makes the CNN network biased towards a certain procedure. (There are 9 milling features compared to 7 turning features that can make the machine side with milling with unsure predictions)
- The training epochs choice and different training affects the output too much.
- The saving and preservation of the CNN model somewhat affects its prediction and results.

In our project we are currently seeking different approaches to improve the results of the predictions before adding new features to our system. Our current trials to improve the results include:

- Feature extraction, to make the number of features equal or block a particular feature that can affect the predictions in a negative way.

- Data extraction to make the number of models of machinable and nonmachinable equal to see if it makes the network biased.
- Use SHAP (SHapley Additive exPlanations) that is a game theoretic approach to explain the outputs of the machine learning systems, to see the weights assigned to different features and their effects on the predictions.
- Create a side project that takes binvox files and improves their details by increasing their resolutions and smoothing the round corners.
- Trials to train the networks with different epochs and parameters, to be able to compare them.

The results of these trials will determine the next steps we will be taking during our project.

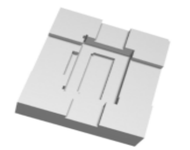
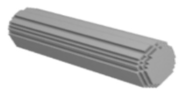
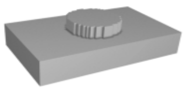
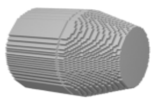
Model	#	Ground truth		1,000 epochs		2,000 epochs		5,000 epochs		20,000 epochs		Data processed		Feature 1 Disabled		Feature 6 & 7 Disabled		500 less milling-turning	
		Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure
	3	Yes	Milling	No	Milling	No	Milling	No	Milling	No	Milling	No	Milling-Turning	No	Milling	No	Milling	No	Milling-Turning
	19	Yes	Turning	No	Milling-Turning	Yes	Milling-Turning	Yes	Milling-Turning	Yes	Milling-Turning	No	Turning	Yes	Milling	Yes	Milling-Turning	Yes	Milling-Turning
	21	Yes	Turning	No	Turning	No	Milling-Turning	No	Milling-Turning	No	Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning
	30	Yes	Turning	No	Milling-Turning	No	Milling-Turning	No	Turning	No	Milling-Turning	No	Turning	No	Turning	No	Turning	No	Milling-Turning
	43	Yes	Milling	Yes	Milling	No	Milling-Turning	Yes	Milling-Turning	Yes	Milling	No	Milling-Turning	Yes	Milling-Turning	Yes	Milling-Turning	No	Milling-Turning
	46	Yes	Turning	No	Milling	No	Milling	No	Milling	Yes	miling	No	Milling	No	Milling	No	Milling	No	Milling
	49	Yes	Milling	Yes	Milling-Turning	No	Milling	No	Milling	Yes	Milling-Turning	No	Milling-Turning	No	Milling	No	Milling	No	Milling-Turning
	51	Yes	Milling	No	Milling	No	Milling	No	Milling	Yes	Milling	No	Milling	No	Milling	No	Milling	No	Milling
Number of correct results / 8				2/8	4/8	1/8	1/8	2/8	4/8	5/8	4/8	0/8	3/8	2/8	4/8	2/8	4/8	1/8	1/8

Figure 17: Test Models - 1 Pink tone is wrong outputs and orange one is discussable, they can both producible milling and turning machining procedures but one procedure is more optimal

1,000 Y + less mil-tur/nonmach		AC 20000 w less mil		AE with all non-mach		S with less milling-turning		using all non-mach data 1000		Feature 8 Disabled		1,000 epochs + Y + less cmd	
Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure	Machinable?	Machining Procedure
Yes	Milling	No	Milling	No	Milling-Turning	No	Milling	No	Milling-Turning	No	Milling	No	Milling
No	Turning	No	Miling	Yes	Milling-Turning	No	Milling-Turning	No	Milling-Turning	Yes	Milling	No	Turning
Yes	Milling-Turning	Yes	Milling-Turning	No	Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Turning
Yes	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning	Yes	Milling-Turning	Yes	Milling-Turning	No	Turning
Yes	Milling-Turning	No	Milling-Turning	Yes	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning
Yes	Milling	Yes	Milling-Turning	Yes	Milling	No	Milling	Yes	Milling-Turning	No	Turning	No	Milling
Yes	Milling	No	Milling-Turning	Yes	Milling	No	Milling-Turning	No	Milling-Turning	Yes	Milling-Turning	No	Milling-Turning
Yes	Turning	No	Milling-Turning	Yes	Milling	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning	No	Milling-Turning
7/8	3/8	1/8	1/8	5/8	3/8	0/8	1/8	2/8	0/8	3/8	3/8	0/8	3/8

Figure 18: Test Models - 2

6 Maintenance Plans and Details

We do not have any plans to maintain this project after we are done with this course since there is no need for maintenance of PeerNews desktop application as we do not have any data server. Anything that is shared via PeerNews is kept on client hosts. Since there is no server, there is no upkeep for that. The only component that requires maintaining is the libraries that are used in the project.

7 Other Project Elements

In this section of the report we will be delving into other elements regarding the project, such as ethics and professional responsibilities and teamwork details.

7.1 Consideration of Various Factors in Engineering Design

The four main factors that we deliberated on are algorithmic efficiency, memory cost, the accuracy of the model, and the time required to predict the attributes for the parts. The first two are common to all software engineering solutions, and we have decided that we would be in favor of algorithmic efficiency as opposed to lowering memory cost, and would make trade-offs in order to speed up our processes even when it means more memory would be required. We have chosen to do so because our training of the model takes quite a bit of time with a large number of epochs and a sizable amount of time per each epoch. Therefore, we have chosen to try and speed this process up as much as we can even though it might mean we would have to utilize more memory to do so. One can argue that because we only need to train the model once before providing it to the user we would not necessarily have to worry about the time it would take to complete the training of the model. However, as during testing we might change qualities about the model in an effort to increase its accuracy, and would have to train it every single time, we have chosen to favor algorithmic efficiency as opposed to a lowered memory cost. One may also argue that since the training has to only be done once the memory cost will only have to be paid once, so it is not a big problem to favor algorithmic efficiency.

7.2 Ethics and Professional Responsibilities

As our project will likely be replacing human jobs in the future, it is our ethical responsibility to ensure that its predictions are at least as accurate as those that an average human worker would make. We also have the professional responsibility of finishing the project on time with all of the functionalities we have detailed present, and of course have the ethical and professional responsibilities to our teammates of doing tasks that we said we would be completing on time and in a decent manner. Throughout the project we will be keeping in contact with our project supervisor and our innovation expert, and we will have the responsibility to ensure that our communication is in a professional and respectful manner.

7.3 Judgments and Impacts to Various Contexts

In this section of the report we will be delving into the judgments and impacts of the project to various contexts in more detail.

7.3.1 Economic Contexts

Our project will be able to automate a number of processes within the creation of a specific part within a given manufacturing plant, and will thus be able to reduce costs for the people involved in the creation of the selected part. This will be in the form of not having to employ people whose roles are to do the processes that will be automated by our project, and will therefore create an economic incentive for people involved in manufacturing and the creation of parts to use our project.

7.3.2 Environmental Contexts

Our project will be able to reduce the number of mistakes or mishaps that may happen during the manufacturing of specific parts. As these mistakes or mishaps result in inefficiencies in the use of the manufacturing plant, which in turn result in the manufacturing plant's use going to waste, generating unnecessary pollution to the environment, our project will be able to benefit the environment by reducing or completely eliminating these mistakes and mishaps.

7.3.3 Social Contexts

Our project will be able to automate some of the processes that are undertaken during the production of specific parts. These automations are likely to either remove the people who were doing the processes which are to be automated from employment or have their efforts focused elsewhere. Thus, our project will have a social impact on the people who are now manually performing the processes which our project is going to be used to automate.

7.3.4 Political Contexts

Our project will be able to be used in large manufacturing plants to increase the efficiency of the plant. As large manufacturing plants are usually used in service of government affiliated companies, our project will be able to in a roundabout way help the government by increasing the efficiency of manufacturing plants affiliated with them. No other political constraints have been linked to our project.

7.3.5 Manufacturability Contexts

Our project will be used to automate some of the processes involved in the manufacturing of certain parts at manufacturing plants. Thus, we will be increasing the efficiency of said manufacturing plants and will as well be reducing the likelihood of mistakes or mishaps from occurring.

7.3.6 Sustainability Contexts

Our project will be increasing the sustainability of manufacturing plants by automating certain processes and thus lowering the risk of mistakes or mishaps from occurring. Our project will also be a sustainable project as the machining operations our project will be based on have been around for many years and are likely to be around for much more. Even if new machining operations are introduced, use of these current machining operations will likely not dwindle as they are core operations, the other machining operations that are to be introduced will likely be used as supplements to these already existing ones. In addition, with sufficient data provided, any new feature can be added to the existing models with additional training therefore the system can stay up-to-date without extra work.

7.3.7 Ethical Contexts

Our project will be used to automate certain processes to be done in the manufacturing of parts. This will likely lead to the people who are doing those jobs manually to be left without anything to do. This will lead to one of two scenarios: either they will be let go and lose their job, or they will be reassigned to do some different type of work. This issue would go into the broader issue of AI replacing menial human jobs, and is an ethical concern. However, those who are in favor of progress such as our group members recognize the fact that such regrettable consequences are part of progress and innovation, and will pursue the project regardless.

7.3.8 Professional Contexts

As stated in the previous constraint, our project is likely to either cause employees to be let go or reassigned, which are the professional constraints associated with our project.

7.4 Teamwork Details

In this section of the report we will be discussing the measures we have taken as a team to ensure we are contributing and functioning effectively as a team, we have created a collaborative and inclusive environment, and have taken leadership roles as well as shared leadership effectively within the team.

7.4.1 Contributing and Functioning Effectively on the Team

We have briefly discussed how we are trying to move the project simultaneously in two activities of development, the first being the actual implementation of the project and the second being the writing of the reports and the documentation of the project within the previous section of the document. This allows us to effectively separate team members and assign them to one or the other of these two activities, ensuring that everyone knows what to do and that they have other team members that they can ask for help if they get stuck. In addition to this, we are having weekly meetings with both only the group members and as well meetings with our project supervisor and innovation expert. During the group member only meetings we are focusing on task allocation for the week and ensuring that everyone has a task that they are to do, everyone knows what this task assigned to them is and that the tasks for the week have been distributed in a fair manner among the group members. In the meetings with the project supervisor and the innovation expert present, we are presenting what we have done during the previous week and will be asking them to give us some feedback on the things that we have done over the previous week. We are also asking for guidance on the direction that the project is going in and whether we are on the right track or not. We have as well created two WhatsApp groups, one with just the project members and the other with the innovation expert and the project supervisor for easier and faster communication on matters that do not require a meeting. This is how we are planning to ensure proper teamwork throughout the project.

7.4.2 Helping Creating a Collaborative and Inclusive Environment

As mentioned previously, we are having weekly meetings both only as the group members and with our innovation expert and project supervisor. This helps create a collaborative and inclusive environment as everyone is able to see each other weekly and discuss what tasks they have been doing throughout the week et cetera. It also helps that every group member knows that there will be a weekly meeting, which makes it so members are more determined to get their work that has been assigned to them done, making it so the weekly meetings serve as a pseudo-deadline for the team members. One other thing is that we have meetings with only the group members present as well as those with our innovation expert and project supervisor present. This helps make it so we can have more informal meetings with just the group members present and more formal meetings with our innovation expert and project supervisor present. The informal meetings allow for faster communication and more rapid assignment of tasks, while the formal meetings help us keep track of where we are under supervision. We also have two separate WhatsApp groups, one with and one without the innovation expert and project supervisor. This helps us communicate more directly and set up meetings easier. We also have a Google Drive in which we keep our resources and work on our reports collaboratively on a Google Docs document. We also have a Discord server with our team members where we can post deadlines and communicate more effectively. We are also using GitHub as a means of working collaboratively on our coding activities. All these tools and services we use as well as our meetings help us create a more collaborative and inclusive working environment.

7.4.3 Taking Lead Role and Sharing Leadership on the Team

We have Irmak as our impromptu leader, as this was her summer internship project before we have picked it up as our senior design project as a team together, therefore she is more knowledgeable when it comes to the terminology or the inner workings of the code et cetera, although throughout the semester we have all more or less caught up to where she is right now. She is also the one that usually contacts the project supervisor and innovation expert to set up meetings as she has known them longer and has had a professional relationship with them prior to us choosing this as our senior design project. However, when it comes to decisions such as how a segment of the code or the report should be written, we discuss the possible alternatives as a team either during one of our meetings or through our WhatsApp group. Every member has an equal say when it comes to these decisions, with those members who are more knowledgeable regarding the topic in which the decision is being made having more of a leadership role. Bulut is our member of the team that is the most experienced in using Python, therefore he has more of a leadership role in the coding side of things. Alp is the member that usually spends the most time on the reports, therefore he is more of a leader when it comes to those. Irmak is more of a leader when it comes to communicating with the project supervisor and innovation experts as well as administrative decisions, Denizhan is more involved with the user interface elements of the project and Tanay is more involved when it comes to the design aspects of our application.

7.4.4 Meeting Objectives

In our project we had several objectives, such as having our model be able to determine whether a part was machinable or not, and then to be able to give us the machining operations that would be required to make that part if it is machinable et cetera. We also had the objective of cost estimation. The first of these objectives was completed near the end of the first term. The second was completed towards the beginning of the first term and the last was completed near the end of the second term.

7.5 New Knowledge Acquired and Applied

For our project, we have thus far had to learn about manufacturing terms such as additive/subtractive manufacturing, methods of manufacturing and machining methods. We were able to learn these concepts with the help of our project supervisors and innovation expert, and as well private study on our own. When required, our project supervisors and innovation expert have sent us some documents that we can study on our own to grasp certain concepts better. This was all work done in order to comprehend the project itself that we would be doing. Regarding the implementation of the project, we will have to learn concepts of machine learning such as convolutional neural networks and modeling, and for that we have members of our group that are taking the machine learning course offered by the university, and we are also doing private study regarding the concepts via individual research. For use of certain machine learning libraries in Python and other pieces of software that we will have to learn, we are reading documentation and trying to find helpful resources online to guide us. We already have a working grasp of GitHub from previous courses such as the Object Oriented Software Engineering course, but will still likely be looking up concepts or operations that we are unfamiliar with online. These are our plans and learning strategies for gathering and absorbing new knowledge.

8 Conclusion and Future Work

This concludes our senior design project. We have aimed and succeeded at creating an ML based system that will input a three dimensional model of a part that wants to be manufactured, and determines whether it is machinable or not, the machining operations to do so if it is, and how much it will cost to do so.

9 References

[1] “What is machine learning?”. https://www.ibm.com/cloud/learn/machine-learning. [Accessed: Oct 9, 2021]

[2] S. G. Joung, V. Aggarwal, J. W. Sutherland, and M. B.-G. Jun, “Identifying manufacturability and machining processes using deep 3D convolutional networks,” *Journal of Manufacturing Processes*, vol. 64, pp. 1336–1348, 2021. M. Szilvśi-Nagy and G. Mátyási,

[3] ”A Comprehensive Guide to Convolutional Neural Networks”. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: Oct 9, 2021]

[4] “Additive vs. Subtractive Manufacturing”. https://formlabs.com/blog/additive-manufacturing-vs-subtractive-manufacturing/. [Accessed: Oct 9, 2021]

[5] “Machining Processes: Turning, Milling, and Drilling”. https://trimantec.com/blogs/t/machining-processes-overview. [Accessed: Oct 9, 2021]