Adam Grimley
B00691778
COM668

Live Doc
Virtual Space Compiler

Contents

1. Introduction

1.1 Project Statement

Through the author's industrial placement, he had worked on immersive technologies. One major proponent of immersive development is the generation of accurately scaled spaces to preserve special awareness for the user. Using game development as a guideline, we can see that development of 3D environments is often one of the largest costs to development [1].

Many programmers - like the author of this report - have been looking at cutting edge research papers and technologies to see if we can shorten this process down in terms of both time and money. It is important to mention that although the author focuses on immersive development, this project is not just an immersive opportunity however it was immersive technology that lead the way to this research so will be mentioned throughout.

Alongside the author's immersive work during industrial placement, he had investigated new ML techniques such as regional convolutional neural networks (RCNN) and pose detection. These techniques have recently presented us (the immersive development community) with a possible opportunity to build a pipeline that will allow us to quickly generate 3D spaces based on a flat photograph of a real space. The components of this pipeline will be discussed and developed throughout the course of this report.

An area of concern for the author through this project will be the scope he is looking at. Although a lot of research has been completed for each section of the pipeline, no one has so far proved they can work together. This is hard to protect against other than making sure each section tests succeed in a vacuum before integrating them. As well as this, each section of the pipe is very recent developments and could be unstable for use in a production environment. In an effort to mitigate against any issues that may arise because of this, frequent correspondence with the authors of the key papers [2], [3] will be attempted through development.

This project makes use of 2 major research libraries which have been recently released. The first is mask-RCNN which is a neural network project which allows us to quickly develop an RCNN which will segment the picture a user has input. This is the component that will allow the product to recognise objects and to isolate them from the picture. This software library has been developed by Matterport and open sourced under the MIT License allowing open use of the software.

The second major component to be used in the project is the recently published KeypointNet project. This software allows for the detection of object pose within the user's image. This section will be placed after mask-RCNN to reduce any error within the detection of each object's relative pose. This research product was developed by Google AI and released under the apache license. The reason the author has chosen this library is to speed up the addition of future objects to recognise. The KeypointNet shows that it does not need user defined key-points to generate pose models. This will be important in the future because we want to make the pipeline as easy to use as is possible.

1.2 Project Aim

The overall aim of this project is to release a way for us to take a single photograph and turn this into an accurate 3D space for use in simulations. The project seeks to, in turn, open more ways for the technology community to hasten the development of accurate 3D locations. The product(s) planned will be an open source pipeline and a research document outlining the end pipeline - assuming success of the project. The pipeline looks to take a photograph and pass this through a trained RCNN and pose detection system hosted in the cloud and return a file which can then be used by the Unity engine to generate a 3D virtual space.

1.3 Objectives

Objectives of the project are outlined in figure 1 below. These objectives will be the general aims of the project and outline what needs to be accomplished at a high level before the author will deem the project a success. Each objective was developed by the author to either test the implementations of cutting-edge machine learning algorithms or test his own knowledge of the subject area.

ID	Objective	Achieved S1	Achieved S1
01	Input a picture from any camera source into system. The initial project scope will use a png image.	N	-
02	UI built within the unity engine	N	-
03	Single object recognition from picture	Y	-
04	Isolate the desired object from a test image	Y	-
05	Recognise the depth of object in relation to the camera	N	-
06	Recognise the global rotation of desired object	N	-
07	Output position, rotation, and depth of object in relation to camera	P	-
08	Use a JSON structure for data transfer between components	Y	-
09	Build API to allow the application to offload processing of image	P	-
10	Use Azure/AWS VM with a public IP for server hosting	Y	-
11	Offload ML (machine learning) model training to the cloud	Y	-
12	Generate 3D scene from JSON structure output by API	N	-
13	Create a selection of models to substitute for objects in 3D scene	N	-
14	Organise system into single application pipeline	N	-

Figure 1 - Objective listing. Assigned completion into Y (yes), N (no), and P (partial)

2. Literature Review

2.1 Part one – Problem domain

To complete this project in a way which furthers the AI (artificial intelligence) community in Northern Ireland and push the boundaries of automated image processing, the author of the project has conducted preliminary research into several topics including R-CNN [3]–[5] and pose detection[2], [6]–[8].

2.1.1 RCNN

Before getting into the reasoning of using an R-CNN within this pipeline, a brief overview of the research that led the author to the conclusion that an R-CNN would be useful. The R-CNN (Region of interests with convolutional neural network features) structure was proposed as an effort to bridge the gap that sat between object classification and detection [9]. The basic R-CNN structure split the process into 3 stages or modules:

- Extract region proposals
- Compute CNN features
- Classify regions

Using an RCNN we essentially generate *r* number of regions of varying size across the picture input. These regions are category-independent. There are many different ways to generate these regions as objectiveness[10], multiscale combinatorial grouping [11], and object proposals [12].

After the initial paper outlining the use of RoIs and CNNs together [9], the next major upgrade to the R-CNN format came in the form of Fast R-CNN (the background of Fast R-CNN is required for the comparisons in figure 3). This methodology proposed higher efficiency of the R-CNN method by implementing feature sharing and multi-task loss during training among other advancements [5]. Researchers developing this method had shown up to 18.3x increased training speed than R-CNN and SPPnet [13], another computer vision method [5]. Unfortunately, Fast R-CNN uses object proposal as a method for determining region of interests in the given picture (RoI). Due to the nature of this project, the author decided it best to look for a different method for determining the RoI rather than manually specifying which object to look for each time the application was executed.

Another route to follow when aiming to fill the object recognition, classification, and isolation was to use the FCN (Fully convolutional network) methodology [14]. Although this is not an R-CNN, it could in theory complete the work required for the object recognition and classification, and image segmentation component of the proposed product. The author's reasoning against using FCN was due to the work required to adapt the methodology to the current program. Although in terms of standalone semantic segmentation FCN could be a great contender, the product theorised by the author would require the instancing of each class. The FCN methodology alone shows to work poorly for this [3] and would require further work to allow this. In the future, once the pipeline is built, there may be a case for transitioning the component into an FCN if experiments show a speed increase.

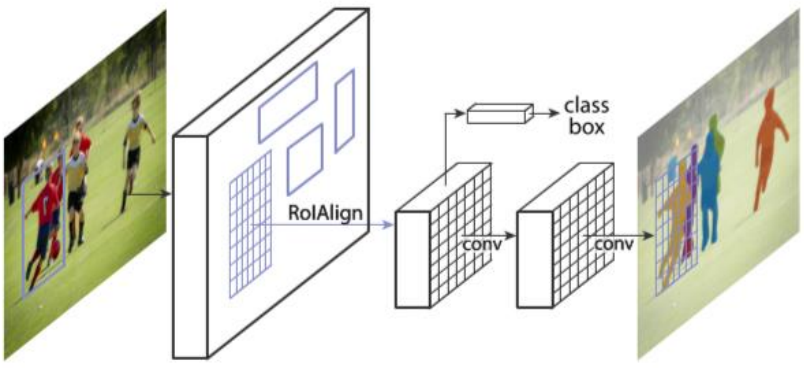

Figure 2 - Mask R-CNN framework [9]

The theorised product would require instance segmentation as previously mentioned. Recently work was published detailing mask R-CNN. This is a variation of an R-CNN with a focus on instance segmentation [3]. This is an extension of the previously mentioned Fast R-CNN and as such is fast to implement and train. This relives time pressures on the Author to implement this component.

In terms of the object recognition component of the product, the system would require the recognition of an object within the input photograph. Once an object is recognised, we would then require the separation of this object from the remainder of the picture on a pixel-by-pixel basis. This combines both semantic segmentation and instance segmentation. The only CNN purpose-built to achieve this task in an efficient manner is the mask R-CNN. Figure 3 below shows mask R-CNN currently outperforms all previously mentioned models. The backbone section of the table outlines the neural network architecture of the model. The backbone is worth mentioning when discussing the accuracy of the bounding boxes because it is the net that learns object patterns.

	Backbone	**Bounding Box AP**
Faster R-CNN+++	ResNet-101-C4	55.7
Faster R-CNN w FPN	ResNet-101-FPN	59.1
Faster R-CNN w G-RMI	Inception-ResNet-v2	55.5
Faster R-CNN w TDM	Inception-ResNet-v2-TDM	57.7
Faster R-CNN, RoIAlign	ResNet-101-FPN	59.6
Mask R-CNN	ResNet-101-FPN	60.3
Mask R-CNN	ResNeXt-101-FPN	62.3

Figure 3 - Comparison between R-CNN models in regard to bounding box accuracy percentile on single models [9]

2.1.2 Pose Detection

In order for the final product to work as expected, we need to not only figure out what the object is with regard to classification (as discussed above) but we also need 3 pieces of contextual information: position, rotation, and depth. In order to achieve this, a pose detection algorithm will need to be put in place. The main issue with pose research is that many of the papers found by the author focused on algorithms to determine human positions where the product would require object poses.

As context for this component regarding the final product, I am only defining the scope to be a selection of two test images. The images will be comprised of 2 different arrangements of 3 objects on a desk. Each object will have a different pose in the images. As the product is focused on objects, plane detection will not be required for this version of the product and as such, a simple plane will be placed at the base of the lowest object detected.

Modern pose detection processes were first presented in 2005 as pictorial structures for object recognition [7], which is still considered a starting point when developing pose recognition systems [15]. The pictorial structures specify a model that will distinguish between instances of an object based on training models. Again, this paper uses human pose to test and validate the algorithms, but it is worth a mention here as although the author does not require human pose, the graphical tree based approach given by the paper may be required knowledge when developing the object pose model.

From the pictorial structures, we can follow pose detection through hierarchical models [6], non-tree models [8], [16], to convolutional models [15] which are commonplace today. The author wishes to use a pose model which allows us to define which way an object sits in the image. For instance: what way the handle of a cup faces as it sits on a table. The convolutional pose detection model would be able to achieve this however there are several downsides the author had found.

The first of these downsides is the keypoint locations. Specifically, the denotation and training of these locations. In the previously mentioned papers, the methods outlined require the manual specification of each key point for each training image. The author believed that there is a better way to undertake training using other software which will automatically find the key points for itself. This would allow for faster training of models and therefore faster addition of more objects to the system. Shortening training time later in development was seen to be a worthy decision even considering the risk of the required upfront development time [Section 4].

The second downside of these models is the output of point depth in relation to the camera. One major issue when developing a system such as the product proposed, is that depth is notoriously hard to gather from a single point perspective for several reasons. Depth cameras often make use of either infrared or stereoscopic cameras to define depth as each of these technologies allows us to make easy distinctions between locations of objects in relation to the camera. The author however, has specified that the product uses single monoscopic images to create the scene which complicates the process. Rather than taking the depth data direct from the camera, we need to instead analyse the context of the object [17].

In relation to the high-level system architecture the author has proposed in section 3.5 of this report, we need to pass the pose components just the masked image theoretically. This proposal may require changing down the line as with a masked object image we remove the global context. We get around this by using recent research into latent key point analysis. This research proves that using a trained model, we can output the depth of key points within the object image [2].

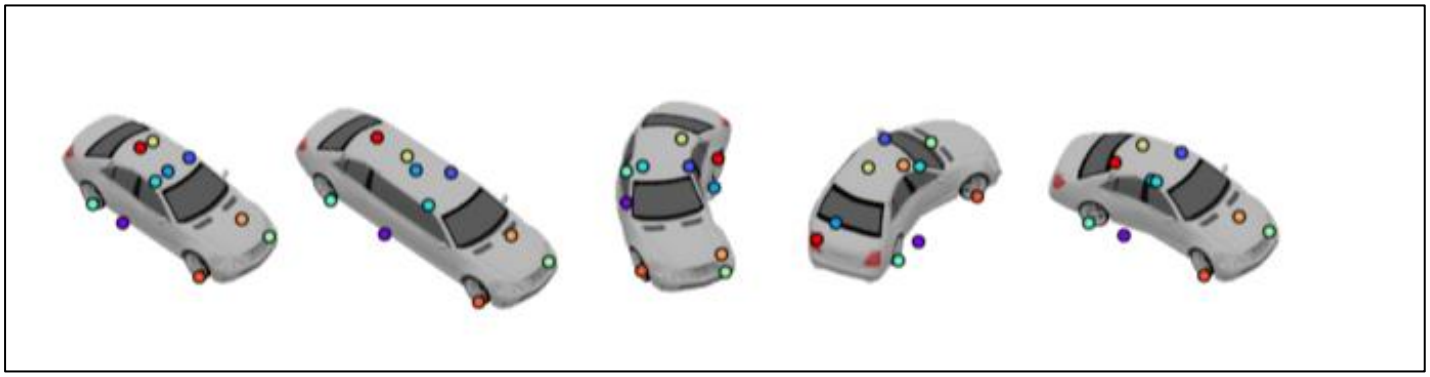

Figure 4 - Keypoints on a deformed car [16]

The main thing that drew the author to this model for depth and pose is the automatic generation of key points. This means that as the system trains, it will define its own key points to use meaning it will always use the most efficient points. The system also analyses and outputs the depth of each key point which can be theoretically used to position the object in relation to the camera. Using this system will reduce the training required for adding objects to the system however does come with a large upfront development cost due to it being experimental technology.

One major risk to using this technology for the process is that it has not been wholly proven against real world scenarios. Throughout the paper [2] the authors have used 3D models to train, test, and validate the model. This may require the author of this project to edit the algorithms used or generate new processes for pose and depth of objects.

2.1.3 Physics engine

To complete this project fully, we will need to have a way to develop an accurate way to simulate realistic physics for our virtual space. As the target for this project is for accurate simulations in virtual reality, this is a key factor in completion of the project to the authors specifications. In researching the physics engine to use, a key factor was the licence type and whether the base software is open source or not.

For this implementation of the proposed product, the stakeholders concluded that a fully accurate simulation is not required, and it will be sufficient to have simply plausible simulation physics. This was a key factor in determining the engine that would be used as an initial development point. It allowed the author to choose an engine in which he could be more comfortable working in without needing too many tweaks.

Through research, it was found that in an accurate simulation, the PhysX engine developed by Nvidia was the most stable open source engine although most engines have their pros and cons [18]. The PhysX engine is also the most accessible for the implantation planned as it is the physics backend for the two major simulation development engines the author has experience using: Unity3D and Unreal. The author decided to use Unity3D for this project due to more experience in advanced simulation development on the platform.

2.2 Part two – Similar Solutions

There are some similar solutions to the problem outlined. In terms of the transferring of a real life space into a 3D scene that can be used in 1:1 scale would be photogrammetry. Photogrammetry is defined by [19] as:

> *"the science of obtaining reliable information about the properties of surfaces and objects without physical contact with the objects, and of measuring and interpreting this information."*

When used in the real world, photogrammetry and LIDAR scanning allows engineers and developers to effectively and realistically scan a given real life area in moments. Matterport, One of the global leaders in photogrammetry products states that their scanners can produce a realistic scan of a $2000ft^2$ area in under 26 minutes [20]. This does not include processing time nor setup time, which increases the total time significantly. These numbers should be taken as anecdotal however, as dealing in scanning the time of the scan depends wildly based on what you need from the location and how many scans you require.

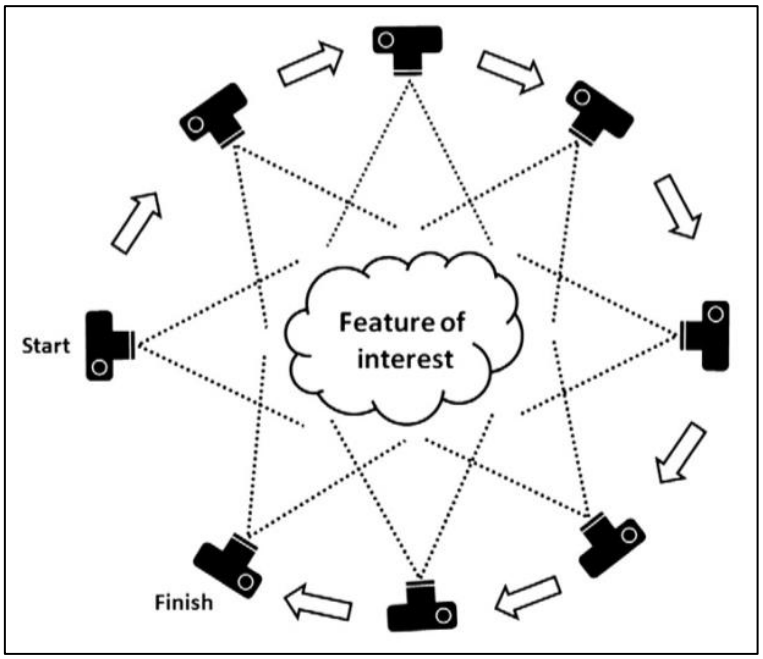

Figure 5 - SfM photogrammetry [21]

The Leica system for LIDAR scanning of an area states a scan time of 3 minutes per single scan [22]. The author has personal knowledge of using this system and understands that a 200ft^2 area from start to usable 3D image can take up to half a day. This scan time – although relatively fast for the actions – is still too long for quick developments and each comes with issues in changing the 3D scan to suit a developer's requirements.

A big difference in the proposed system would be the use of custom models in substitution of real-world textured models. Where a 3D area would be generated based on the textures and measurements from the real world, the proposed system removes the real-world textures. Our system uses the measurements and locations of objects and theoretically, would generate the area faster because of this.

The systems do however have different applications. A geospatial scan such as a scan from LIDAR or generated using photogrammetry have their uses is realistic representations of areas. This is a use case that is key to preserving history and spaces for the future. The proposed system of this report would be used to generate quick 3D spaces using custom models based off a single image. There may be a use case down the line where, using feature recognition or location mapping we could scan and generate full 3D spaces. At the present, due to the research nature of the project, there is only the scope for a single image to be used meaning a restriction in the space generated.

A system such as the product proposed would be used to prototype 3D systems in a realistic space quickly and efficiently. The author came to the idea of this project from developing immersive systems and not being able to test in a real-world scenario without weeks of work beforehand. The system could also be used as a soft segmentation tool to 'touch-up' any 3D scans with professionally developed 3D assets.

In researching the applications that are similar to the proposed project, the author had found a recently released paper documenting 3D-RCNN, a convolutional neural network with the purpose of distinguishing 3D features from a 2D image [4]. This works in a similar way to the proposed system however the end product is to be used for different cases. The system proposed by [7] is a real-time system for generating a 3D shape set that can be used for depth and analysis. This is key for autonomous vehicles and other systems that require accurate and fast depth images from 2D graphics.

The system proposed in this report uses object recognition to substitute specified objects in a 2D image with 3D assets in a virtual space. This means that developers may use custom objects to quickly create realistic 3D spaces based off real-world scenarios. Developers using this system are not locked to recreate the real scenario completely.

It is a testament to the research nature of the proposed project that the 3D-RCNN work is not yet published. The paper published is an outline of the algorithms used and how it is built but there is not functioning system so far. Although the use cases are different, the area of work is the same between the author's proposed system and 3D-RCNN.

3. Project plan and Requirements Specification

This section outlines the initial project planning stages as well as justification for requirements gathering methodology and initial specification. As the product is a work in progress, some of the initial requirements may change through the product lifecycle.

3.1 Stakeholder Identification

This project has 2 distinct categories of stakeholders: direct and indirect. The direct stakeholders are those closest to the project where the indirect are possible end users. The proposed project, however, is research focused and as such the main goal of the project is to further the development of 2D image recognition technology using recent advances in the field of machine learning. It is due to this, that the main stakeholder listed is the author of this report. All other stakeholders are indirect. These are listed in figure 6 below.

Stakeholder ID	Name	Role
D01	Adam Grimley	Developer / Author
I01	Dr Gaye Lightbody	Mentor / Advisor
I02	Kainos Innovation Team	Advisors
I03	PSG	Peer support / Focus group

Figure 6 - Stakeholder listing

If required by the author, specialist machine learning developers (as outlined in section 1.1 of this report) may be consulted. The author does not intend for this to be required however if they are consulted, they will be added as an indirect stakeholder in a consultant role.

3.2 Requirements gathering methodology

In gathering requirements, a meeting was placed with members of the Kainos Applied Innovation team to discuss what would be required of a system like that which has been proposed. The team are recognised members of the Northern Ireland research and development community and as such, have experience in projects such as this. The author conducted this meeting using carefully selected questions designed to create a discussion around the proposed application and to identify and analyse any unforeseen “edge cases” that may come from standard use of the eventual product.

Going forward in the development of the application and research, the author has gathered a focus group to further develop requirements and to receive feedback about the product. This has shown to be a non-discriminatory way of increasing productive input during the development of a product [23] and has been chosen by the author for this reason. These meetings will give people a safe environment to raise concerns about the product direction and to suggest ways in which the product can be improved.

From these focus group meetings, the suggested changes will be taken on board, checked to see if they can fit into the project scope, and passed through as a change request. These requests are examined by the author and checked against the schedule of development to see if they are valuable and time efficient. These may also be discussed with the author’s mentor to examine whether the changes will add a tangible increase in the project viability. Each requirement listed above has been taken from the initial meetings with Kainos and the focus group and may change or be added to as the project is in development.

3.3 Requirements prioritisation strategy used

After being introduced to the technique in university, the author decided the best way to prioritise requirements would be to use Karl Wiegers Relative Weighting, evaluating each requirement based on cost, value and risk [24]. In the case of the proposed project, the cost value will represent the estimated time for each requirement to be fulfilled. The main component of the prioritisation strategy is the equation:

Value percentage / (cost percentage * cost weight) + (risk percentage * risk weight)

This equation allows us to prioritise each requirement based on weights of how important the cost or the risk is. Due to the high risk of this project, the risk weight is set slightly higher than the cost as the author believes time to be less of a factor than the technical complexity. An example of the relative weighting used is in figure 6 below.

Relative Weights:	**1.0**	**1.0**			**1.0**		**1.2**		
	Relative Importance	**Relative Penalty**	**Total Value**	**Value %**	**Relative Cost**	**Cost %**	**Relative Risk**	**Risk %**	**Priority**
Totals	122	98	220	100	86	100.000	75	100.000	
Application is standalone	9	6	15	6.8	2	2.326	2	2.667	1.234
Application takes single photograph as input	9	8	17	7.7	2	2.326	1	1.333	1.968
user does not process the image locally	7	5	12	5.5	7	8.140	8	10.667	0.260

Figure 7 - Requirements prioritisation example

This process shows which requirement should be handled first. In this case, it would be the single input functionality.

3.4 Initial Requirements

Requirements listed in the table below are split into FR (functional requirement) and NFR (non-functional requirement) IDs and linked to the objectives listed in section 1.3 of this report.

ID	FR/NFR	Requirement	Priority
01	FR	Application is standalone and does not require a browser	1.234
02	FR	Application takes single photograph as input	1.968
03	FR	User does not process the image locally	0.260
04	FR	Develop a public endpoint server that allows connection to ML models from anywhere	0.409
05	FR	System shall be trained using common objects	0.451
06	FR	System will hold standard 3D models of common objects	0.536
07	FR	System will recognise and report the relative rotation of desired object	0.450
08	FR	System will recognise and report the depth of desired object relative to viewpoint	0.350

09	FR	The system will automatically communicate between different components of the pipeline	0.352
10	FR	The system will substitute 3D models in virtual space based on where the specified objects are in the input image	0.256
11	FR	The user should be able to show/hide metrics during execution	0.579
12	FR	The user should not have to leave the application during the process	0.829
13	FR	The application should have Virtual Reality support	0.096
14	FR	The application should allow the selection of objects to create in the 3D space	0.680
15	NFR	System will be developed using cutting edge open source frameworks for ML	0.176
16	NFR	System will have a way to report metrics including stage times, accuracy percentiles, etc.	0.536
17	NFR	System will not hold any data that isn't marked as training data	0.987
18	NFR	System shall have a minimum of 70% test coverage	0.581
19	NFR	The API that receives and returns data should be secure	0.736
20	NFR	The application should be desktop only	1.152

Figure 8 - Initial requirements

3.5 Proposed System Architecture

As an example of the architecture of the system going forward in planning, the author has outlined below how the system will be structured.

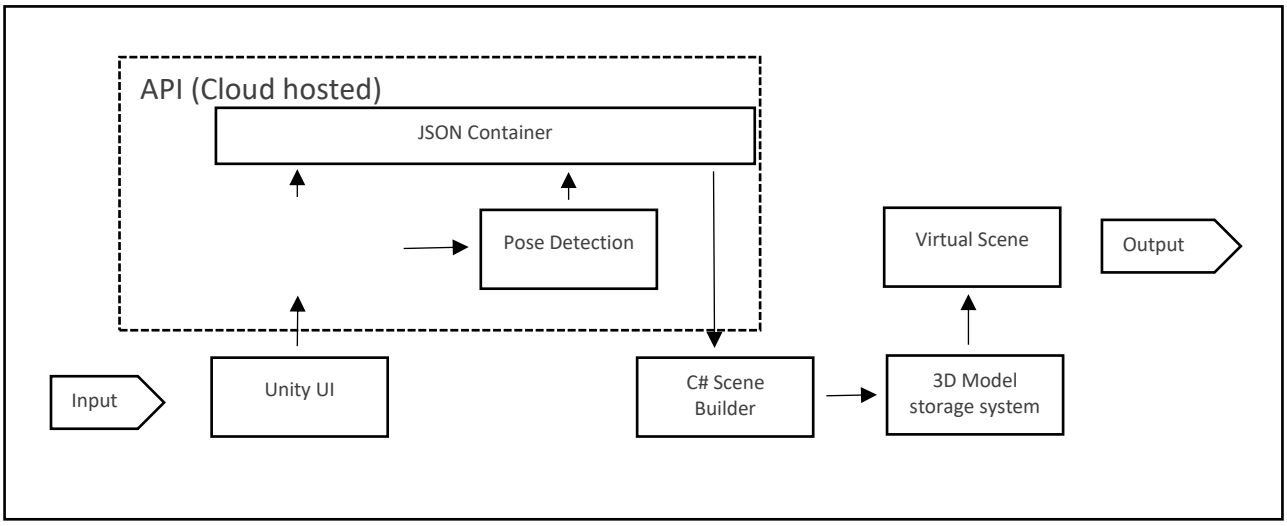

Figure 9 - Proposed Architecture

3.6 Software Lifecycle Methodology

As outlined throughout this report, the full system proposed is comprised of many different components. Each component should in theory be able to work independent from the others however it is the connections between each component that make the system novel. Even so, because of the independent nature of each component, the author has decided that rapid prototyping will be the best practice going forward, so that each component can be prototyped and connected from the ground up in relatively similar timeframes.

It is because of this decision, that the methodology chosen for the project is spiral development. This methodology was first introduced by Boehm in 1988 combining waterfall and RP development [25]. The justification of this is the timeframes of the project set by the university. These given timeframes lend themselves more naturally to a waterfall driven development life cycle which the author believes would not be efficient for a single person development team to implement into this project.

The spiral methodology gives more forgiveness to complex systems [25], such as the system proposed in this report, and allows for faster reaction to unforeseen technical complexity. This is required in the project as some of the technologies that will be used are still experimental and not fully tested. The methodology proceeds as cycles of development, at the end of which there will be a product prototype. At the beginning of the cycle the author will evaluate risks associated and react accordingly – developing the highest risk components first.

In association with the spiral methodology, a Kanban board will also be used. This will be held online and hosted by GitKracken. The Kanban board will be used for tracking the progress of each development cycle. The proposed cycles are listed in figure 10 below. As the project is still in the planning stages, these rounds may vary slightly in the later stages of development in keeping with the methodology.

Cycle	Task ID	Effort	Description
1			**Preparation**
	01	L	Proposal
	02	H	Initial project plan and report
	03	M	Research into component technologies
	04	M	Requirements development meetings
2			**Basic API**
	05	L	Setup cloud VM with public access
	06	H	Train RCNN model with single object
	07	H	Train Pose model with single object
	08	M	Move models into the cloud
	09	L	Connect models using JSON container
3			**Basic Unity integration**
	10	L	Build initial UI
	11	L	Integrate the posting of inputs to the VM public IP
	12	M	Build initial Scene builder to parse returned JSON
4			**Model Building**
	13	M	Build initial simple object recognised by API
	14	M	Generate structure to hold models
	15	M	Integrate structures with scene builder component to spawn objects
5			**Additive Development**
	16+	M	Train API with additional model
	17+	M	Create appropriate 3D model
	18+	M	Integrate model with scene builder component

Figure 10 - Cycles outline

In figure 10 above, cycle 5 will continue to be repeated in the remaining time as in theory all systems will be connected by this cycle and should just need added to. The author only expects one of these cycles to be completed however there is space for continued addition if the time is available.

Each cycle will begin with a meeting with stakeholders to outline and examine any risks associated and will end with all systems tested to assigned standards.

3.7 Work Breakdown Structure

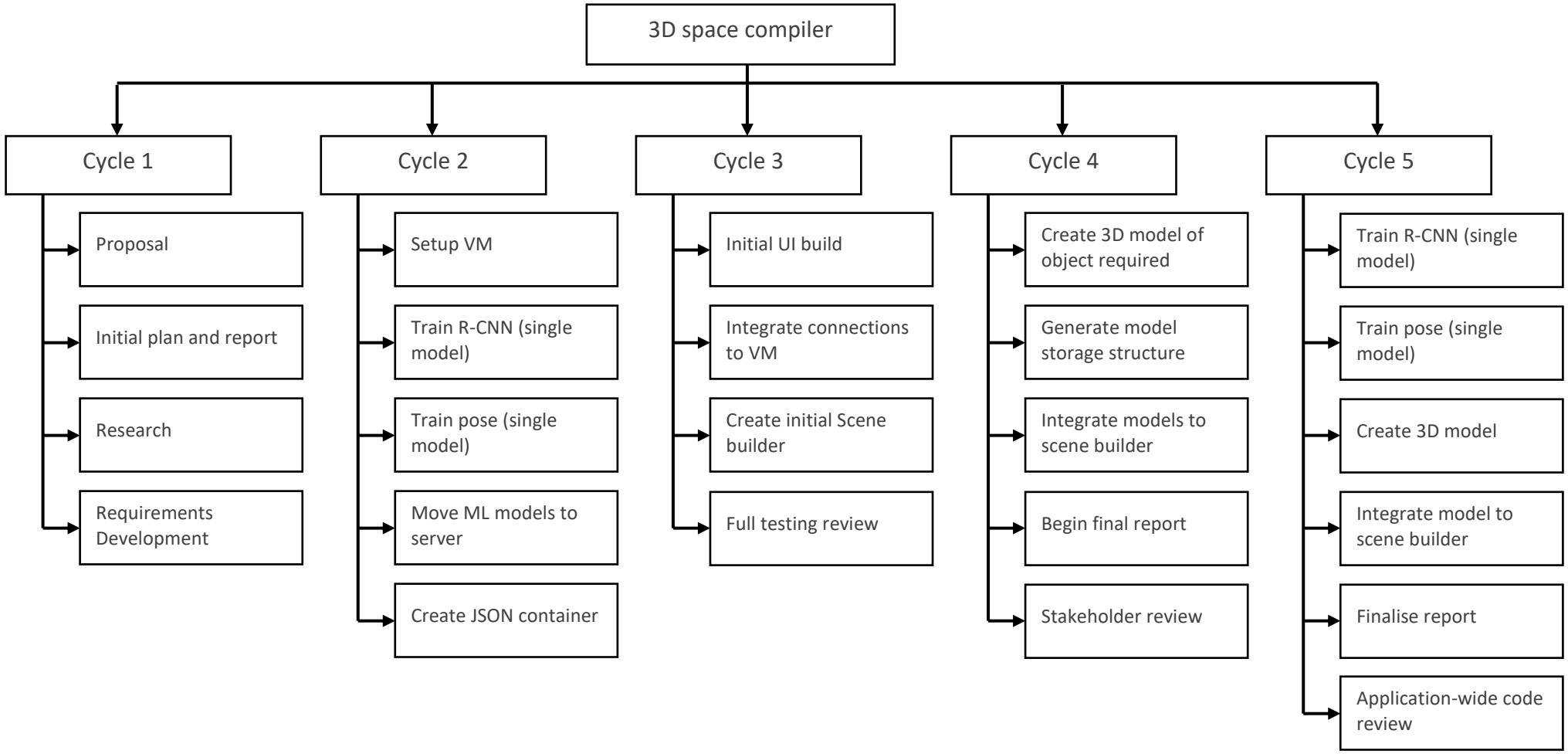

Figure 11 - Work breakdown structure

3.8 Gantt Chart

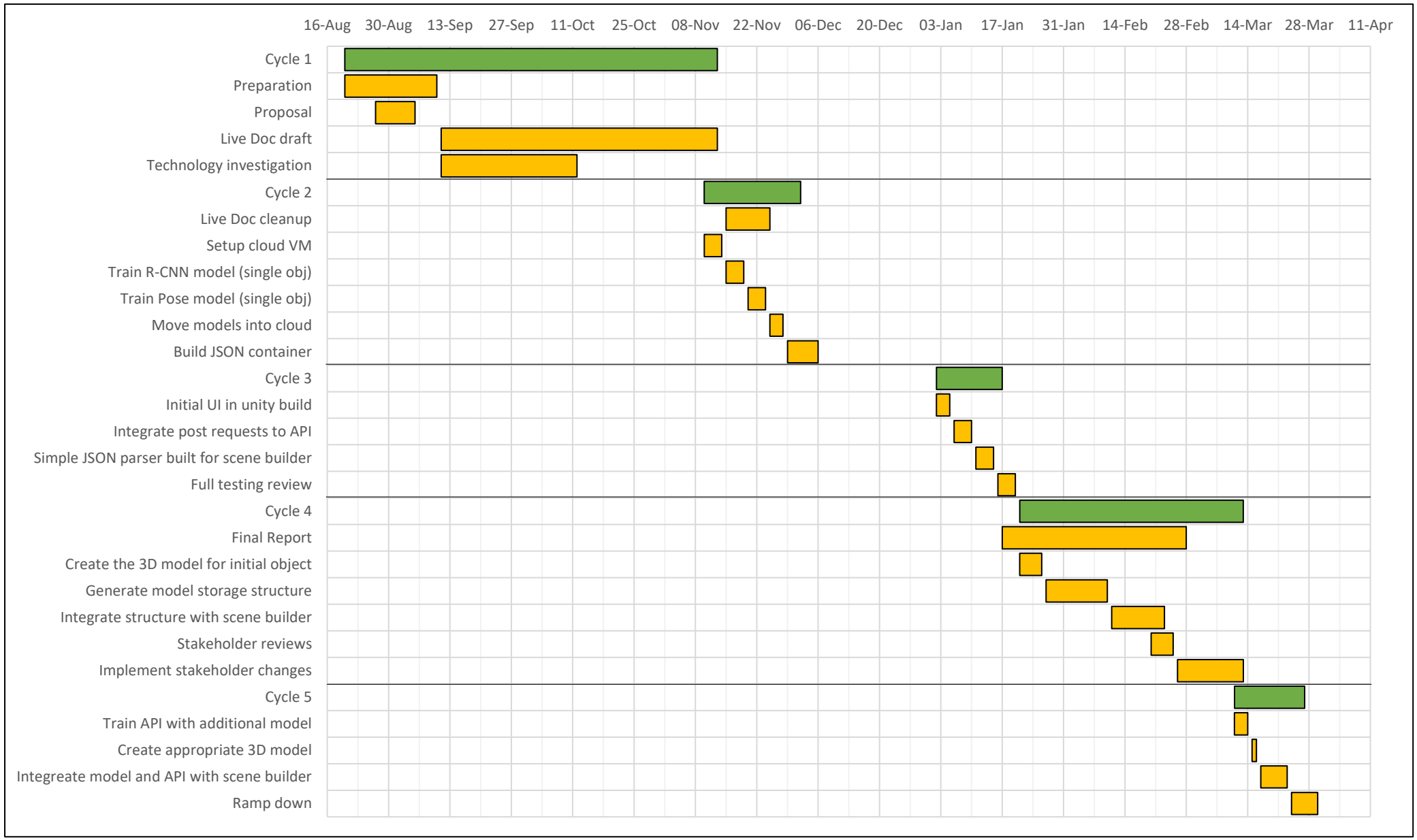

Figure 12- Gantt Chart. Tasks in yellow, cycles marked in green. Each task is represented from figure 10.

3.9 Resources identification

Resource	Type	Purpose
Laptop	Hardware	Main form of development. Main piece of hardware for the project and will be used throughout.
AWS	Infrastructure	Will be used to host the python API in the cloud. Also will be used to train ML models as a VM cluster will train faster than a standard laptop.
Jupyter Notebooks	Software	Python notebook development environment. All python will initially be developed within jupyter as it allows easy and readable modularisation of python files.
Git	Software	Will be used for source control throughout the project.
Visual Studio 2018	Software	Used for C# development. Is provided by Microsoft Imagine and will hook directly into the source control.
Git Kracken	Software	Online software allowing quick generation of detailed Kanban boards. This will be used to track the cycles of development and any changes that are requested.
Unity Engine	Software	Physics engine that will be used to generate all visuals and 3D scenes. Will also call to the API when required.

Figure 13 - Resources

Due to the project only being developed by a single engineer, human resources have been omitted from identification. This is also because the project is not funded, and these costs are negligible.

3.10 Verification plan

The project aims to make use of 2 completely different types of ML models. This adds considerable technical risk as neither have been designed to work together. Each of the specified models will be tested separately using industry standard 80/20 split of the data [26]. This means that each dataset while training the ML models will be split into 80% training data and 20% verification data. Once each model is tested individually, developers will then continue the testing development and start to look at the actual code of the program.

In testing the inner code of the program, I plan to use an automated workflow. These tests will be broken into unit tests, integration tests, compatibility tests, and user acceptance tests. The use of automation in the tests will provide the developer with the ability to cover all these areas and quickly see if changes break anything already developed and pushed in a previous development cycle.

Throughout the development of both python-based components and C# based components, unit tests will be developed before the full development of the component. This ensures the efficient programming of modules using a TDD methodology.

At the end of each cycle, the current prototype will be put through black box testing. The results of these tests will play a part in defining the work required in the coming work

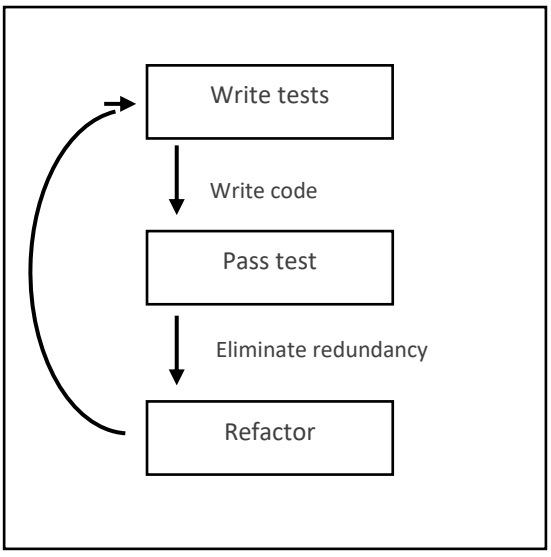

Figure 14 - TDD diagram

cycle, as outlined in the spiral development methodology [25]. Each black box test will use both the proposed test images as inputs. A verification test for each requirement is listed below.

Requirement ID	Verification test	Pass criteria
01	Run application on a clean install	Application does not require any other software to be installed
02	Follow application through to image submission	Application only asks for a single image
03	Submit an image to the system for analysis	Application does not request more resources than are required by a regular unity development session.
04	Connect to the system from a new IP address	Application allows instant analysis regardless of new IP
05	When going through image analysis, open object selection window	Application lists several common objects approved by stakeholders
06	Build a scene using the system	Application automatically uses standard 3D objects with no input required from the user
07	Analyse an image with the system	Application returns a list of object rotations
08	Analyse an image with the system	Application returns a list of depth points of the desired object
09	Run application from start to finish	Application does not require any input once image is submitted
10	Run application from start to finish	Application generates accurate virtual space based on user input
11	In the main screen, select to view application metrics	While application is running, it will show metrics on the screen rather than blank loading screen
12	Run application start to finish	Application stays within a single window throughout
13	Build a scene using the system	Once scene is built, user can put on a VR headset automatically
14	On the main screen, open the object selection pane	Objects will be listed
15	-	-
16	Run system from start to finish with log option selected	A log folder will be generated on user's desktop
17	-	-
18	Run test coverage program in the backend	Application has 70% or higher coverage
19	-	-
20	Try run application on a mobile tablet	Application does not run

Figure 15 - Verification plan

3.11 Validation Plan

The proposed product will be validated through regular meetings with stakeholders. These stakeholder meetings will ideally be held at the end of each cycle and will address any issues that come from the current prototype at the time. All stakeholder will be asked to individually discuss each task outlined in the previous cycle and how the final prototype reflects each. These discussions will produce work items that will be added to the next cycle if the project lead believes there to be enough time for development of these changes before the next major stakeholder review.

4. Risk Assessment

The initial risk table is outlined below. This table outlines the criteria of risk, the threat posed by the risk, source of the risk, the control measure to reduce risk likelihood, then the risk scores generated by the author. The final column also outlines the mitigation strategy of the risk. This strategy is the plan of action to take should the risk happen. The risk score is generated using: *Likelihood * Impact = Risk.*

ID	Criteria	Threat	Source	Control measures	Likelihood	Impact	Risk	Mitigation strategy
01	Technical	Not enough resources to train models efficiently locally	Local CPU	Use AWS vouchers to move model training to the cloud. Voucher sources either github student or AINI community.	6	4	24	If the vouchers prove not to be enough, contacts from both Kainos and Digital Catapult will be able to supply computers with top of the line GPUs for model training.
02	Technical	Classification of objects with R-CNN not accurate enough	ML Model training	Spend more time developing a larger training and validation dataset for the R-CNN component	3	7	21	Revisit the training dataset and see if you can improve upon the tagged images. If the dataset is deemed large enough by the project lead, research a different image classification model.
03	Technical	KeypointNet system does not return the true depth of each object	ML Model	N/A	6	8	48	If the KeypointNet system does not return the true depth, look into monoscopic depth perception algorithms such as [17] and estimate how long it would take to implement.
04	Technical	VM server could drop due to connection issues	Cloud provider	Using AWS to host the VM server will allow the risk to be almost non-existent.	1	3	3	Host the server locally.
05	Technical	The JSON data may not translate into appropriate unity scenes	API	This will be mostly a risk with the model data outputs. Care will be taken when training and developing the models to be used such that we return the correct data	2	7	14	Edit the model codebase to allow access to the correct information.
06	Technical	Unity cannot send the images to the server in the correct format	Physics Engine	The use of a image converter library should allow the translation of an image into a base64 string. This string should be accepted by all formats.	1	2	2	Use a different encoding algorithm on the hosted API
07	Human	Scope of project may be too large to deliver	Specification / Requirements	The scope of the project should be revisited at the beginning of each sprint and risks mitigated. The scope of the project is large however with continual revaluation, all objectives can be met	7	5	35	The scope should be reduced if it becomes intangible. A one-page document should be produced confirming this and outlining the reasons for the scope reduction.

Figure 16 - Risk assessment table

To summarise the previous risk table, the highest risks come from the ML models to be used and the overall scope of the project. The author of this report has already deemed the project to be large in scope and understands this risk. This risk is one of the reasons for the spiral life cycle methodology and has been outlined throughout the planning section this report. The technical complexity comes from the use of experimental systems that are being used and tested. These systems are only recently published and have not been proven in production environments as far as the author has researched. Extensive research has been carried out concerning these systems and the author has concluded that – although it is an untested environment – both systems will be able to perform the task within the product with great efficiency.

Medium risks come in the form of data outputs and training datasets. Each of these risks have low likelihood yet high impact on the project should they appear. They are both easily mitigated if they appear, with one only requiring a larger dataset and the other requiring the editing of system codebase. The main impact of these is that they will be time consuming to deal with but not complex (based on preliminary research carried out by the author). They also each have the risk of unknowns. The code of each model used is largely abstracted and may not provide us with the correct data we need. If the training dataset is greater than 100 images total for each, the object chosen may be too generic for the system to correctly distinguish from an image. If this happens, a new dataset will need to be created for a new image which is heavily time consuming and will cause missed deadlines.

The remaining risks have been deemed low risk to the project. These are outlined in green on figure 16. These risks, although they will have an impact, either have an extremely low likelihood or are easily mitigated against.

5. Initial Prototype

5.1 Risk and rationale

In developing the initial prototype as the first major submission and milestone of work product, the author decided to focus on a risk that will allow him to hit as many major objectives as possible. With this in mind, the author has decided to develop around the classification of objects and the accuracy associated with this. This risk is listed as a medium risk to the project above.

The reason behind this was that it was deemed an achievable risk to mitigate against if it did go wrong. The submission assigned to this project by the customer (the university) would not leave enough time to fully develop a deliverable should something go wrong with risk 03 in the time outlined after necessary research in the planning section of this document. This is due to the technical complexity of the KeypointNet library and the experimental nature of the training process. Instead, the author has chosen to focus on the R-CNN system because he has a very limited prior knowledge of the library and believes he could mitigate against issues in an easier and more direct way.

Along with this reasoning, the R-CNN system is designed as the first section of the API. This reduces on the data the author may have needed to mock to build any other piece of the API. It also allows the beginning of development of an API and hosting this on a publicly accessible location. Furthermore, it would give him a perfect starting point for building the JSON structure that will be carried throughout the proposed system.

Risk ID	Description
01	Not enough power to train models efficiently locally
02	Classification of objects with R-CNN not accurate enough
04	VM server could drop due to connection issues

Figure 17 - Risks developed in prototype

In summary, the author has chosen to take a medium risk as the focus point of the prototype over a high-risk factor because it allowed me to mitigate and develop through several other risks listed. A high-risk factor is still being tested in the form of risk 01.

5.2 Design Artefacts

5.2.1 JSON structure

In planning for the prototype, the author first needed to outline a structure in which the system could pass data efficiently from one component to another. This comprised of what would be passed to the API through a post request, and what would be passed throughout and output from the API. These structures are listed below.

```
{
  “image”:(base64 encoded image),
  “objects”:(object list to detect),
  “training flag”:(bool value)
}
```

Figure 18 - Data to be passed to API

```
{
  "image":(base64 encoded image),
  "RoI":[(list of RoIs in image)],
  "cropped images":[
     {
          "cropped image":(numpy array)
          "saved filename":(string),
          "location":(string),
          "rotation":[(float list)],
          "object":(string)
     },
     {…}
  ]
}
```

Figure 19 - JSON that will be output from API

Each of these JSON structures may change slightly later in development however, for the initial prototype they contain all data the system will need.

5.2.2 Training images

In training the R-CNN model to recognise cups the author was required to hand annotate images of cups. This is a time-consuming process using an online tool called VIA image annotator. The annotations come back as a json file which can be used by the mask R-CNN training functions. Some of the training images and their annotations are listed below. In total, 15 images were annotated in this way, with a 12/3 split in images between training and validation.

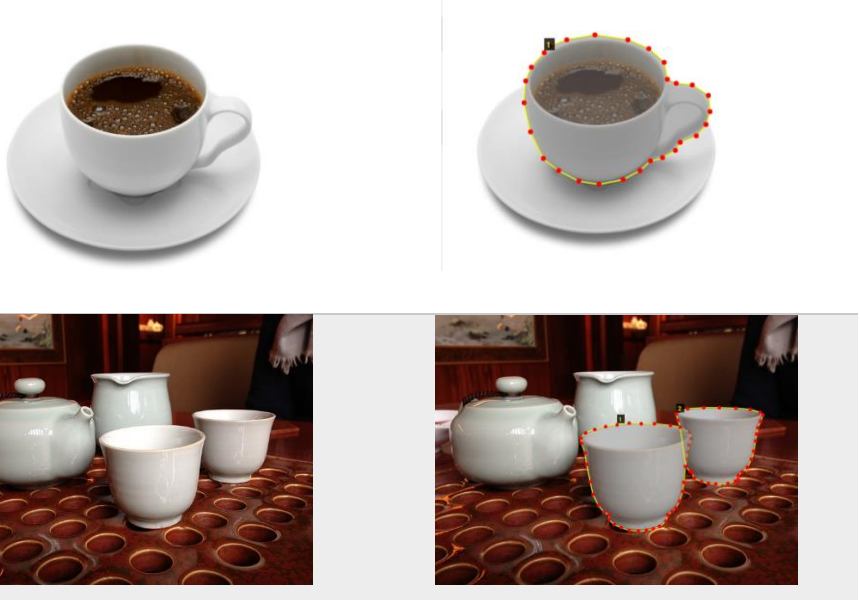

Figure 20 - Training/Validation images and annotations

5.2.3 Prototype dataflow

The prototype has a relatively simple dataflow, however, it tests many different pieces of the infrastructure as well as testing different methods of training the required ML models. The dataflow is outlined in figure 21 below.

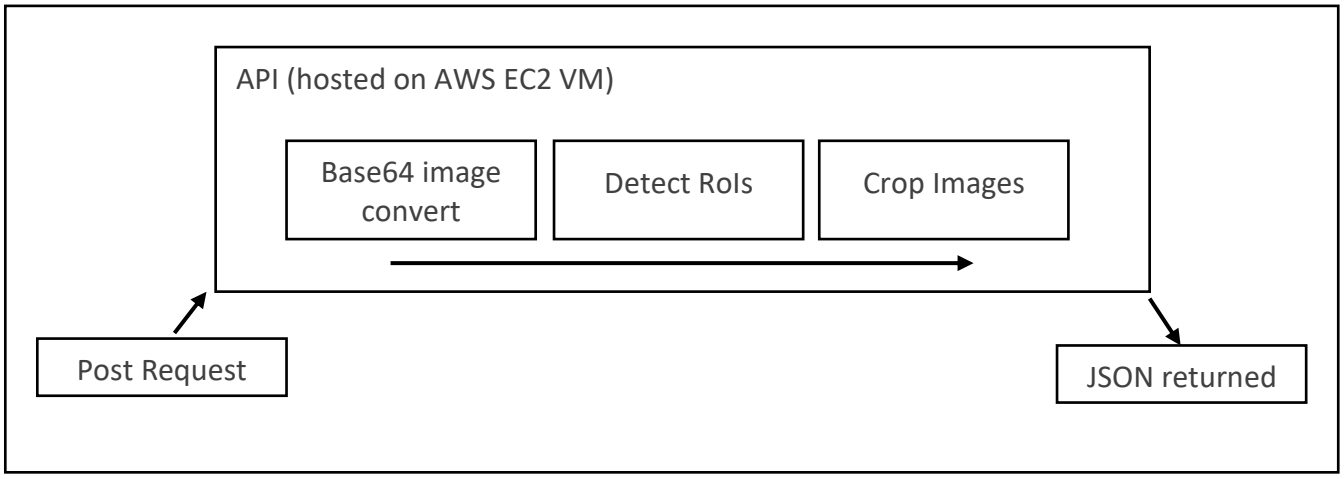

Figure 21 - Prototype dataflow

5.3 Issues in prototype development

As was to be expected at this stage of development, some issues were encountered when developing the initial prototype. The first major hurdle to this part of the project was the time it took from beginning of development to understanding the how to fully use the mask R-CNN library. This turned out to require a week's work for the developer to develop a working knowledge of tensorflow, matplotlib, and the R-CNN library itself. This is something the author did not foresee during planning and unfortunately because of this, the project is now one week behind schedule.

Another issue that arose was the training of the model itself. The author had outlined this in the risk assessment and was a major risk he was testing with the prototype. The author acted out the mitigation plan that was in place and after noticing that the VM he had setup for the API was not able to use compute resources (more on this later), he transferred learning onto my local machine. Through this machine the author was able to pool more resources and was able to train the R-CNN model with the training dataset. A large issue with this was that while training, he was not able to complete work on his local machine. Depending on the training time this would not normally be an issue however, the model's estimated training time per epoch was 2.5 hours. In completing 17 epochs, the local machine was unable to be worked on for 2 days.

This training time did not introduce unforeseen issues due to planning on the author's part for this. It does however point to what will be a larger problem going forward where there is more pressure to have as much uptime on the local machine as possible. To mitigate this, the developer took 3 days to setup a separate EC2 instance that allows for better GPU compute resources. In addition, it increased the VM RAM from 1GB to 61GB, and 1 available core to 4. This dropped epoch train times to 12-25 minutes even when poorly optimized. More development time will be focused on optimizing training configurations to make full use of the new instance going forward.

The final hurdle to the prototype was the API instance in the cloud. When everything was complete and the code was ready to be moved into the VM, it was found that the VM did not have enough memory to complete an analysis on a single image. Due to the nature of EC2 instances, the author could have upgraded the memory at cost. Instead, he looked for a cheaper solution at this point in development to avoid large personal cost. He came across a technique called swap files that allow the allocation of additional memory at the cost of performance. This performance drop is caused by the offloading of memory to the hard disk. The performance drop is acceptable at this point in the development. A larger instance will be used for the final product demonstration.

6. References

[1] A. Grossman, *Postmortems from game developers*. 2004.

[2] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi, "Discovery of Latent 3D Keypoints via End-to-end Geometric Reasoning," no. 1, pp. 1–13, 2018.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017–Octob, pp. 2980–2988.

[4] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare," *Cvpr*, 2018.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 1440–1448.

[6] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.

[7] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, 2005.

[8] L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2012, pp. 2–9.

[10] B. Alexe, S. Member, and T. Deselaers, "Measuring the Objectness of Image Windows," vol. 34, no. 11, pp. 2189–2202, 2012.

[11] P. Arbel, J. T. Barron, and U. Polit, "Multiscale Combinatorial Grouping," vol. 500.

[12] I. Endres and D. Hoiem, "Category Independent Object Proposals."

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[14] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[15] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[16] X. Lan and D. P. Huttenlocher, "Beyond trees: Common-factor models for 2D human pose recovery," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005.

[17] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," pp. 1–9, 2014.

[18] J. Hummel, R. Wolff, T. Stein, A. Gerndt, and T. Kuhlen, "An evaluation of open source physics engines for use in virtual reality assembly simulations," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[19] T. Schenk, "Introduction to Photogrammetry," 2005.

[20] "How long does it take to scan a property?," *support.matterport.com*, 2018. [Online]. Available: https://support.matterport.com/hc/en-us/articles/229136307-How-long-does-it-take-to-scan-a-property-. [Accessed: 26-Oct-2018].

[21] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, "'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, 2012.

[22] Leica, "Leica blk360 imaging scanner." Leica Geosystems, p. 1, 2017.

[23] J. Kitzinger, "Qualitative Research: Introducing focus groups," *BMJ*, 1995.

[24] K. E. Wiegers, "Software Requirements, Second Edition," *Microsoft Press*. 2003.

[25] B. Boehm, "A Spiral model of software development and enhancement," in *Software Management, Seventh Edition*, 2007.

[26] A. Geron, *Hands-on MachineLearning with Scikit-Learn & Tensorflow*. 2015.