# 1. Introduction

## 1.1 Project Statement

Through my industrial placement I 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 the largest costs to development [1].

Many programmers like myself 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. I should mention that although I had and continue to focus 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 my immersive work during placement, I 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 me through this project will be the scope I am looking at. Although a lot of research and examples have been done 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 [CORRESPONDENCE WITH DEVELOPERS].

This project makes use of 2 major research products 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 uses. This is the component that will allow the product to recognise objects and to isolate them from the picture. This software component 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 product 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 I have chosen this product is to speed up the addition of future objects to recognise. The KeypointNet shows that it does not need user defined keypoints 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 will in turn open more ways for the technology community to hasten the development of accurate 3D locations. The end project will be open sourced and released alongside a research document outlining the end pipeline. The pipeline looks to take a photograph and pass this through an RCNN and tensorflow 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

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

# 2. Literature Review

The proposed project from the author states that using a system of interconnected ML techniques we can now translate a single, monoscopic photograph into a fully realised 3D space. This comes from the author’s background in immersive technologies wherein the generation of accurate 3D spaces is a time consuming part of development [1]. As such, this is a key point in which the author believes that can be improved to become more efficient.

The author has proposed the use of two key ML technologies that have been developed recently:

* R-CNN
* Pose detection

Both of the above technologies are still recent and are constantly evolving. The first of the two (R-CNN) is a system to use convolutional neural networks to recognise objects in a photograph through semantic segmentation [2]. This system has evolved several times in the previous few years to become a powerful tool for software engineers.

The second technology listed is a technology that is under constant development. Pose detection is the use of ML models to identify the “pose” ie. The position and rotation of the object(s). This technology is often focused on human poses

# 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, just like all others has 2 kinds 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 recognition technology using recent advances in the field of machine learning. It is because of this, that the main stakeholder listed is the author of this report. All other stakeholders are indirect. These are listed in the table 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 3 - Stakeholders

If required by the author, discussions with specialist machine learning developers as listed 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 NI 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 examine any unforeseen edge cases that may come from standard use of the 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 [3] 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 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 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 [4]. 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 4 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 | 17 | 22 | 39 | 100 | 19 | 100.000 | 13 | 100.000 |  |
| User does not process the image locally | 4 | 6 | 10 | 25.6 | 5 | 26.316 | 3 | 23.077 | 0.475 |
| System will hold standard 3D models of common objects | 5 | 7 | 12 | 30.8 | 8 | 42.105 | 2 | 15.385 | 0.508 |
| System will recognise and report the relative rotation of desired object | 8 | 9 | 17 | 43.6 | 6 | 31.579 | 8 | 61.538 | 0.413 |

Figure 4 - Example of Relative Weights

This process shows which requirement should be handled first. In this case, it would be the 3D models. Although they have a low risk, the time required to generate both the models and the data structure required is very high according to the author.

## 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 |
| 01 | FR | Application is standalone and does not require a browser |
| 02 | FR | Application takes single photograph as input |
| 03 | FR | User does not process the image locally |
| 04 | FR | Develop a public endpoint server that allows connection to ML models from anywhere |
| 05 | FR | System shall be trained using common objects |
| 06 | FR | System will hold standard 3D models of common objects |
| 07 | FR | System will recognise and report the relative rotation of desired object |
| 08 | FR | System will recognise and report the depth of desired object relative to viewpoint |
| 09 | FR | The system will automatically communicate between different components of the pipeline |
| 10 | FR | The system will substitute 3D models in virtual space based on where the specified objects are in the input image |
| 11 | FR | The user should be able to show/hide metrics during execution |
| 12 | FR | The user should not have to leave the application during the process |
| 13 | FR | The application should have Virtual Reality support |
| 14 | FR | The application should allow the selection of objects to create in the 3D space |
| 15 | NFR | System will be developed using cutting edge open source frameworks for ML |
| 16 | NFR | System will have a way to report metrics including stage times, accuracy percentiles, etc. |
| 17 | NFR | System will not hold any data that isn’t marked as training data |
| 18 | NFR | System shall have a minimum of 70% test coverage |
| 19 | NFR | The API that receives and returns data should be secure |
| 20 | NFR | The application should be desktop only |

## 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.

Input

Unity UI

R-CNN

API (Cloud hosted)

JSON Container

Pose Detection

C# Scene Builder

Virtual Scene

3D Model storage system

Output

Figure 5 - Proposed Architecture

## 3.6 Software Lifecycle Methodology

As outlined above and 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 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. 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 would not be as efficient for a single person team to implement into this project.

The spiral methodology gives more forgiveness to complex systems [5] such as that proposed 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 round. The proposed rounds are listed in figure 6 below. As the project is still in the planning stages, these rounds may vary slightly in the later stages of development.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 6 - Cycles outline

In figure 6 above, cycle 5 will continue to be repeated 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 Gantt Chart

Figure 7- Gantt Chart. Tasks in yellow, cycles marked in green. Each task is represented from figure 6.

## 3.8 Resources identification

|  |  |  |
| --- | --- | --- |
| Resource | Type | Purpose |
| Laptop | Hardware | Main form of development. Main piece of hardware for the project and will be used throughout. |
| Azure Cloud | Software | 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 8 - 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.

# 4. Risk Assessment

# 5. Initial Prototype

# 6. References

To complete this project in a way which furthers the AI community in NI and push the boundaries of automated image processing, the author must research further into the area of RCNN development and pose detection with a focus on objects rather than human pose. The result of this research will allow a more in-depth customisation of the ML components in the project.

Each area outlined above will be broken into relevant subsections below documenting current research into modern methodologies. This research has been completed before the beginning of the project and may be subject to change due to the fast-moving nature of these technologies.

### 2.1.1 RCNN

The first proposal for an R-CNN (Regions with CNN features) structure was an effort to bridge the gap that sat between object classification and detection [2]. The basic R-CNN structure proposed 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[6], multiscale combinatorial grouping [7], and object proposals [8].

After the initial paper outlining the use of RoIs and CNNs together, the next major upgrade to the R-CNN format came in the form of Fast R-CNN. This methodology proposed higher efficiency of the R-CNN method by implementing feature sharing and multi-task loss during training among other advancements [9]. Researchers developing this method had shown up to 18.3x increased training speed than R-CNN and SPPnet [10], another computer vision method [9]. Unfortunately, Fast R-CNN uses object proposal as a method for determining RoIs. Due to the nature of this project, the student 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 [11]. This is not an R-CNN, however could fill the same section for image recognition. The author’s reasoning against using FCN factored in the manipulation 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 [12] 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.

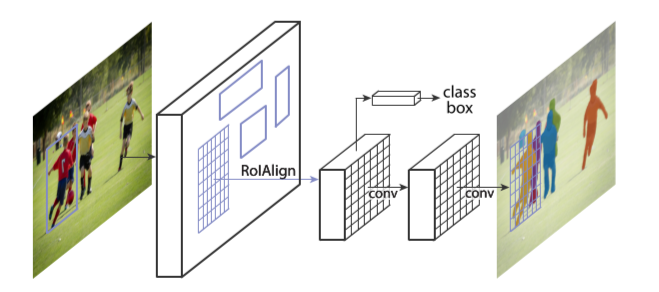
The theorised product would require instance segmentation as previously mentioned. Recently work was published detailing mask R-CNN. This is a variation of an R-CNN with a focus on instance segmentation [12]. This is an extension of the previously mentioned Fast R-CNN and as such is fast to implement and train. This relives time pressures on the Author to implement this component.

Figure 1 - Mask R-CNN framework [9]

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 2 below shows mask R-CNN currently outperforms all previously mentioned models.

|  |  |  |
| --- | --- | --- |
|  | 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 2 - Comparison between R-CNN models in regards 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 pose algorithms focus on human positions where the product would require object poses. This adds a lot of complexity and time to train as traditionally each pose marker would need to be manually placed in a set of training images for the system.