[Date]

12ShiroY

AI Computer Vision

OCR Computer Science Project

Table of Contents

[Analysis 1](#_Toc517781149)

[The Problem 1](#_Toc517781150)

[The Stakeholders 1](#_Toc517781151)

[Existing Solutions 1](#_Toc517781152)

[The Essential Features 1](#_Toc517781153)

[Potential Limitations 2](#_Toc517781154)

[Hardware and Software Requirements 2](#_Toc517781155)

[Success Criteria 2](#_Toc517781156)

[Design 4](#_Toc517781157)

[Decomposition 4](#_Toc517781158)

[Structure 4](#_Toc517781159)

[Algorithms 4](#_Toc517781160)

[Usability 4](#_Toc517781161)

[Variables, Data Structures and Classes 4](#_Toc517781162)

[Iterative Development 4](#_Toc517781163)

[Further Data 4](#_Toc517781164)

[Developing 4](#_Toc517781165)

[Iterative Developments 4](#_Toc517781166)

[Prototypes 4](#_Toc517781167)

[Structures 4](#_Toc517781168)

[Examples of Code (Annotated) 4](#_Toc517781169)

[Variable Names 4](#_Toc517781170)

[Validations 4](#_Toc517781171)

[Review 4](#_Toc517781172)

[Evaluation 4](#_Toc517781173)

[Success Criteria 4](#_Toc517781174)

[Development In Future 4](#_Toc517781175)

[Future Improvements 4](#_Toc517781176)

[Usability 4](#_Toc517781177)

[Bibliography 4](#_Toc517781178)

# Analysis

## The Problem

Above all preference, I would like to create an AI with the ability to find an object in its field of view, which is necessary to find. I will most likely use a bright red stress ball as seen bellow, as this has a very clear and bright colour and a very specific spherical shape, making the object unique, allowing me to decrease the room for error in the computer vision. I will do this by linking a Pi-Camera to a raspberry pi, and creating a program, using public libraries online, to achieve this.

## The Stakeholders 1.1

Currently, my stakeholders are people who are interested in computer vision, who would like to see this similar idea be used to track any ball within sports. This could make life much easier for camera crews in sports in the future, and much more cost effective as this eliminates a significant amount of skilled human work, which could be replaced by artificial intelligence. My stakeholder age range is between the ages of 14 and 40, as this can range from students interested in sports and/ or film.

My main stakeholder is Tom Smith, who is a 15-year-old male, with an interest in both computer science and Sports. He would like to be able to have a camera, which tracks his tennis ball, while he is playing. Tom would also like to have this possibly implemented with an ability to record himself playing tennis, and possibly create content for a platform such as YouTube. The convenience of having a stakeholder similar to Tom is that he can provide very quick and accurate feedback frequently, allowing me to understand the possibility for improvement. This will allow me to build upon multiple prototypes and have a relevant member of the public to test my product from a separate view to me. This could allow me to make my hardware and/or software more ergonomic to the user.

This was the response from my stakeholder, Tom Smith, when I asked him 10 questions about the project.

1. What age range do you think this project is suitable for?  
   I would like it to be suitable for 15-16 year olds like me. Either gender
2. Do you find anything confusing about the idea for the project?  
   No.
3. Do you find anything confusing about the hardware?  
   No
4. Do you find anything confusing about the software?  
   I am not sure yet what the software would run on - that is for you to decide. As long as it can track the ball, I do not really mind.
5. Would you buy this product from a store?  
   Yeah, if it was relatively cheap and looked high quality. It'd be creating YouTube videos etc.
6. What would you want this product to be able to do?  
   It must be able to track a ball, hopefully a tennis ball, so I guess it should sense where the ball is in the field of view and track it. If it could learn from its mistakes that'd be great.
7. Do you see any potential application of this in real life?  
   Yeah, I see great potential for this project in the sports industry and potentially in the film industry, as this could save extreme amounts of money for companies spent on human resources.
8. How simple do you think that the user interface needs to be?  
   The user interface should be simple to use and must include things such as begin tracking; display a message when the ball is located in the field of view.
9. Would you like to use a voice operated system for this product or not?  
   That would be awesome!

## Existing Solutions 1.1

Hawk-Eye, developed by Paul Hawkins, is a computer system owned by Sony, used in many sports to visually track the trajectory of the ball and display a profile of its statistically most likely path as a moving image. This piece of technology is used for sports such as tennis, football, cricket and more. In summary, this is six very high performance cameras used to create a 3D representation of the ball’s trajectory, frame by frame. This then allows the use of complex mathematics to predict the movement of the ball based on its trail. It is not perfect, but it is pretty close, as Hawk-Eye is accurate to 3.6 millimetres.

### In-Depth Breakdown of Hawk-Eye

All information learnt, and screen shots used are from this video (“Visual tracking of a tennis ball” - <https://youtu.be/iRlWw8GD0xc>) on the 13/11/2018.

Assuming that ball candidates (what has been detected as a potential tennis ball) in each frame have already been detected, tracking the tennis ball is then broken down into which candidates, are object-originated (true positives), and which are clutter-originated (false positives).   
The candidates can be plotted in a row-column-time 3D space. The objective of this is to recover the class labels of the candidates. The approach used here is “a layered data association scheme.” A candidate triplet (three frames containing the potential tennis ball in each of them, very close to one another) is selected from the beginning of the sequence, which has an extremely high probability of continuing with only true positives. A dynamic model is then fitted, and then optimised recursively until convergence (the other candidates (near the triplet) seem to line up). This optimised model is called a “tracklet”. As a sliding window moves, a sequence of tracklets are generated. Simultaneously, a graph is constructed, where each node is a tracklet, and the edge difference between two nodes is defined according to the “compatibility” of the two tracklets. The graph is then sectioned into sub-graphs, and the optimal path in each graph is found. The desired data association result is then contained in the optimal paths. Next, interpolation and hit/bounce detection is used to create the final product, the tennis ball tracking computer vision.



## The Essential Features 1.1

The essential features of this project consist of an AI which races against a human opponent on command and easy to use software which is gender neutral. This is the heart of the project so without this being achieved, the whole idea collapses. I will try to focus my attention on these features in order to complete my task as soon as possible.

## Potential Limitations 1.1

The price of this project is a major limitation as in the worst case scenario; I may have to spend over £100 on this project. However, I will try to avoid this by using second hand items such as a Scalextric track and self-made alternate hardware such as a homemade Google AIY kit. Another limitation is the complexity of the coding necessary to complete this task. This project requires an extremely high knowledge of python which I do not yet possess and this will mean that I will have to put in extra time at home to build my knowledge of this. Another possible limitation for the project would be if there was a flaw with the hardware (e.g. response time or frame rate of camera, faulty pins on the raspberry pi etc.). This could potentially limit the processing speed and overall, limit the potential speed of the AI. I will try to avoid this by keeping track of the computer usage throughout the evolution of the project, making sure nothing gets overloaded or potentially break, which could spawn other limitations such as price instantly.

## Hardware and Software Requirements 1.1

|  |  |  |
| --- | --- | --- |
| Number of Requirement | Hardware Requirements | Software Requirements |
|  | A raspberry pi – most likely a raspberry Pi zero as I am working on a limited budget but ideally, a Raspberry Pi 3 B+, as it has a variety of essential upgrades in comparison to the cheaper alternative. I would also need to buy essential parts to use the raspberry pi such as their power block, a micro SD card and potentially, a case (to prevent any damage being done to the raspberry pi). The features of the Pi Zero: BCM 2835 SOC @ 1GHz, 512MB of RAM, micro-SD, mini-HDMI, micro-B USB for data, micro-B USB for power, CSI camera connector (needs adaptor cable for an extra £5), Unpopulated 40-pin GPIO connector, Compatible with existing HAT add-ons, Dimensions: 65mm x 30mm x 5mm  The Features of the Pi B+: 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU (BCM2837), 1GB RAM (LPDDR2 SDRAM), On-board wireless LAN - dual-band 802.11 b/g/n/ac (CYW43455), On-board Bluetooth 4.2 HS low-energy (BLE) (CYW43455), 4 x USB 2.0 ports, 300Mbit/s ethernet, 40 GPIO pins, Full size HDMI 1.3a port, Combined 3.5mm analog audio and composite video jack, Camera interface (CSI), Display interface (DSI), microSD slot, VideoCore IV multimedia/3D graphics core @ 400MHz/300MHz | AI – an evolutionary AI which learns how to move the car around the track, when it crosses the finish line and how to decrease the time taken to complete a lap. This AI will do this using computer vision and measuring the outputs via ammeter and voltmeter. |
|  | (possibly) Voice Recognition kit for Google Cloud Speech – this consists of 170 tie point mini breadboard, 10 male to male 20cm breadboard cables, MAX9812 amplified microphone, NPN transistor, red LED and two resistors (1k and 220ohm). | A UI – easy to use UI (ideally voice controlled) but else if, an easy to use program on the computer |
|  | Scalextric track and cars– most likely donated to me by a friend, however if the track he has is too outdated, I may need to invest into this myself, which would limit me further on the budget for this project. | Faulty computer vision – if the computer vision cannot see where the car is on the track and where the start/finish line is. This could cause major issues with my project and could take a significant of time to fix. |
|  | Voltmeter and ammeter – in order to measure and regulate electrical currents outputted to the Scalextric track | Currently no other software limitations in 1.1 |

## Success Criteria 1.1

|  |  |  |
| --- | --- | --- |
| Criteria | Explanation of Criteria | How Criteria is Met |
| 1) There must be a start and finish line | The program must be able to recognise the start and the finish line so that it can understand whether or not it has completed a lap | I will have red tape on the beginning/ end of the track in order for the camera to see if the Scalextric car crosses the finish line. This will be done via computer vision |
| 2) The program needs to understand where the car is on the track | The program must understand at all times whether the car is on the track or where on the track the car is. | I will implement this by placing green tape on the car and using computer vision for the program to understand whether the car is on the track and/ or where on the track the car is. |
| 3) The program needs to control the car | The program needs to be able to control the car, changing the speed of it when it reaches bends and straights. | I will implement this by using a volt-meter and an ammeter to measure and release electrical currents, controlling the car’s movement. I will also use a neural network for the program to learn how to move across the track with an optimal speed. |
| 4) The track needs to be in the shape of a figure of 8 | The track needs to be a simple figure of 8, as suggested by the stakeholder, built from working Scalextric pieces which still work. | This will be done by using second hand pieces for the Scalextric track, which will be tested by me |
| 5) The whole project needs to be suitable for both genders | The project needs to be simple, easy to use and suitable for both genders | I will implement this by keeping by project as easy to understand as possible for someone without any knowledge of computing. I shall also keep the colours and style based around brown and black (general “boxy” and “homemade” aesthetic). |
| 6) The project needs to be suitable for ages between 15-16 | The project needs to be ergonomic for users and suitable for a wide variety of children/ teenagers of different ages | I will do this by keeping my product slick, easy to use and exciting. This will be achieved by not undermining the potential which this technology can achieve. |
| 7) (possibly) The program will use voice command | The program needs to use voice command in order for the user to turn the program on | I will do this by using the google AIY kit which comes with a microphone and basic voice commands already preinstalled. These can be altered to be specified to my specific situation |

# Design

## Decomposition

My project breaks down into 2 main sections, software and hardware. These then break down further as shown in the hierarchy diagram above. This shows the basic “barebones” concept of my project, which shows how I could potentially dedicate time to each section, building it up to form the combined final product.

## Structure

## Algorithms

## Usability

## Variables, Data Structures and Classes

## Iterative Development

## Further Data

# Developing

## Iterative Developments

## Prototypes

## Structures

## Examples of Code (Annotated)

## Variable Names

|  |  |  |
| --- | --- | --- |
| Variable Name | Variable Type | What The Variable Does |
| CarSeen | Boolean | If the car is seen by the program then, it sends a message to display this |
| Current | Numeric | Stores the value of the current |
| FinishLine | Boolean | If the finish line is seen by the program then, it sends a message to display this. |
| LapTime | Numeric | Stores value of time from beginning of the lap, to the end. |
| BestLapTime | Numeric | Stores the value of the best time completed by the AI and attempts to outmatch it with every evolution. |

## Validations

## Review

# Evaluation

## Success Criteria

## Potential In Future

## Future Improvements

## Usability

# Bibliography