Waterford Institute of
Technology School of Science

Basketball Shot Detection with Deep Learning

Waterford Institute *of* Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Emmanuel Sedicol

20072377

Supervisor: Pat McInerny

A report submitted in partial fulfilment of the requirements
for the degree of
Bachelor of Science (Honours) in Applied Computing

in the

Department of Computer and Mathematics

Panel Reader 1: Pat McInerny

Declaration

All sentences or passages in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Signature:

Name:

Date:

Acknowledgments

I would like to give a special mention to all my lecturers over the last three years, my course leader and Mr. Pat McInerny who has provided me the knowledge and guidance I needed throughout this research phase of my final year project, I thank you all.

I would like to thank my classmates, Alex, Loti and Kacper for their valuable suggestions and comments about my project. I couldn't have asked for better colleagues to share this unforgettable experience with.

To everyone that participated in this project, thank you for your time and co-operation.

Last but certainly not the least, I sincerely acknowledge the efforts of my noble parents for their constant moral support and sacrifice to give me the best possible education.

Abstract

In general, basketball is all about athletes gaining competitive advantage over other players and win matches by enhancing individual capabilities to improve team performance. Traditionally with the guidance of a personal instructor athletes train to improve self-flaws, but how does the trainer know which flaws to work on? With the availability of big data and increase in computational power the instructor is able to analyse the athlete's shooting percentage in previous games and uses the data gathered to determine what a player needs to improve.

This project aims to build a web application that can enhance a user's shooting ability and an application that is free of use to everyone. The web application will be able to process a static video and outputs a statistical view of the users shot attempts.

This paper presents multiple different approach to deep learning and an understanding to how computers are able to have a high-level accuracy of detecting objects. Clear explanations on the process of frame extraction from a static video taken by a mobile camera and different neural architecture that is available today. Understand the difference between using Python or MATLAB before attempting any machine learning projects and get a brief overview of object detection frameworks such a YOLO and R-CNN.

.

Table of Contents

1 Introduction 5
1.1 Background 6
1.2 Aims and Objectives 9
2 Methodology 10
2.1 Participants 11
2.2 Project Materials 11
2.2.1 MATLAB vs Python 11
2.2.2 Camera 12
2.2.3 Database 12
2.2.4 YOLO Object Detection 13
2.2.5 R-CNN Object Detection 13
2.3 Build Design 14
2.3.1 Shot Attempt Classification 15
2.3.2 Basketball Detection 15
HSV Color Map 16
Shape Contour and Diameter Extraction 16
2.3.3 3D to 2D Conversion 17
..... 17
2.3.4 Mathematical Computation 18
3 Prototype 18
4 Discussion and Conclusion 19
4.2 Limitations 19
4.3 Improvements 20
4.4 Conclusion 20
Bibliography 21

1 Introduction

In general, basketball is all about athletes gaining competitive advantage over the opposing team and win matches by enhancing individual capabilities for better team performance. With the availability of statistical data in sports and intelligent applications instructors or coaches are able to analyse the total skill percentage of an athlete based on data from previous games. The data gathered is then used to determine what a player needs to improve next.

Nowadays data mining in sports is a luxury that almost every athlete takes advantage on in order to evolve their level of game play. This paper is mainly focused on the different methods and technologies needed in building an intelligent tracking system to help athletes visualized their shooting ability. Using object detection with deep learning, the tracking system takes in a video input and outputs a statistical representation for every shot attempt. In short, the application calculates shot percentages using python and uses convolutional neural networks to detect the athlete frame by frame.

1.1 Background

Artificial Intelligence or AI for short is a technique that enables a machine to mimic human behaviour. AI was first introduced to us in the year 1951 where Minsky and Dean Edmunds built the very first artificial neural network using three thousand vacuum tubes in order to develop a network of forty neurons. Today AI has evolved into a broader umbrella where the topics machine learning and deep learning fall as a subfield of AI. (Press, 2016)

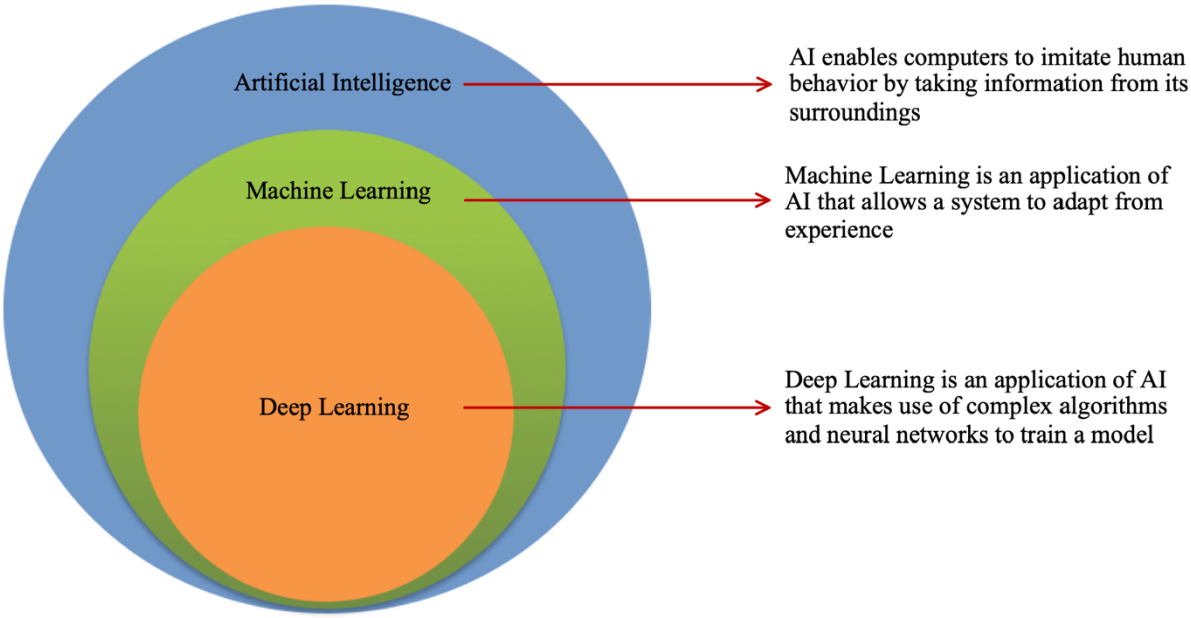

Figure 1.1: *AI and its inner applications*

Deep learning in short is an approach to achieve machine learning by training artificial neural networks using either one of the two concrete methods called supervised learning and unsupervised learning. Supervised learning trains the neural network using a labelled data, while unsupervised learning trains the neural network using unlabelled data and searches for recurring patters. This research mainly focuses on the use of convolutional neural networks but overall there are a total of four major neural network architectures: (Goodfellow, Bengio, & Courville, 2015)

- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks
- Recursive Neural Networks
- Unsupervised Pretrained Networks (UPNs)

In multiple computer vision objectives Convolutional Neural Networks acts as the basic building block for object detection due to its efficiency in classifying which class the detected object belongs to in terms of its pixel density. In context of deep learning CNN algorithm works the best mainly because in most images, objects have consistency in relative pixels which can be a major input to the convolution layers. Deep into the CNN architecture, computer vision tackles the following issues: (Gibson & Patterson, 2017)

1. Object Classification: classifying every pixel of an image and labelling them according to the class they belong to.
2. Object Localization: learning which class an object belongs to and locate its position.
3. Multiple Object Detection: detecting more than one object on a single image and classifying which class each objects belongs to.

With deep learning requiring algorithms to observe and provide accurate results of a big data set, a flexible and easy to use programming language is needed. This is where Python comes into place. Python is a flexible platform-independent language that is compatible with platforms such as Windows, LINUX and macOS. Python having many reliable open source library, this enables programmers to reduce development time and fix bugs efficiently. Overall Python offer various unique technology stacks and libraries for machine learning such as: (Gupta, 2019)

- NumPy offer n-dimensional array for data analysis and manipulation
- Pandas offer numerical tables and series for data analysis and manipulation
- Keras, TensorFlow, and Scikit-learn for machine learning
- Seaborn for visualizing data based on matplotlib

1.2 Aims and Objectives

Basketball being one of the top rising sports worldwide the aim of conducting this project is to enable start-up athletes or anyone who love to play basketball enhance their shooting ability. To achieve this, we aim to build a web application using Python with the following functionalities:

- extract video frames as an image from input video using Python's open source library called OpenCV.
- using the extracted frames locate the exact position of the athlete whenever he or she shoots the basketball and display the located position on a 2D basketball court image.
- display a statistical representation of the overall shot attempts, clearly distinguishing the made shots and the missed shots using coloured markers.

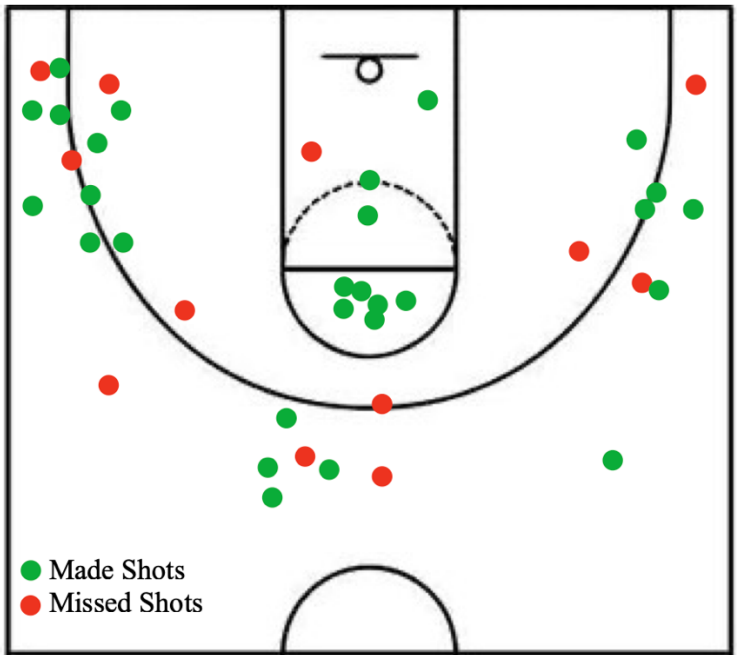

Figure 1.2: *Location of every shot attempts*

Deep learning and neural networks have the reputation of providing an inaccurate probability and estimates, which results to an adversarial example. To avoid from suffering to such error, we aim dig deeper into various detection algorithms such as YOLO object detection to determined which algorithm is the most accurate. To further increase the accuracy of object detection two objectives were laid down. First objective is to investigate on how image filtering can affect either the detection of the basketball or the athlete. Second objective is to investigate about the use MATLAB as an alternative computing environment to Python and state the difference in performance and flexibility.

By the end of the first semester the main aim is to be able to build a small prototype that is capable of detecting an orange coloured basketball. This will take a small video file and display the x-coordinate and y-coordinate of the detected basketball on the terminal screen. When the prototype is fully completed in a shorter time period an algorithm that will accurately calculate the diameter of the basketball will be next on the job list. The diameter will help determine whether the basketball went in the hoop or not.

2 Methodology

In this section, the methodology of the research that was carried out for this semester in order to reach the desired functionalities will be discussed. The selection of software used for building the prototype and the constraints of using either a phone camera or a digital camera will be explained in this section.

2.1 Participants

The participants that took part in my prototype were my old coach and a basketball athlete name Jack. The video is five seconds long which consisted of Jack shooting the basketball behind the three-point arc, clearly showing a perfect shooting form. Video was taken at a single angle so that the prototype is able to detect the basketball quicker due to less camera movement making the ball more visible.

2.2 Project Materials

2.2.1 MATLAB vs Python

Over the years computing has grown into a much bigger community with people continuously evolving the use of languages such as Python and MATLAB. Nowadays when it comes to analytical and quantitative computing Python is at the top compared to MATLAB mainly because of its unique libraries, straightforward syntax and a huge community support. MATLAB is an array-based programming language that also offers similar scientific libraries similar to Python. Libraries include data visualization and image processing which is similar to matplotlib and OpenCV in Python.

In deep learning the execution time for an algorithm is a crucial factor specially in situations where execution is real time. Based on the experience of working with both languages I have decided to go forward with the project using Python. Main reason is that MATLAB is not great when implemented into a web application and a license need to be purchase for every MATLAB toolbox before use. Python on the other hand offers a free web application framework called Django and all its library can be installed for free on the command prompt.

2.2.2 Camera

The initial goal for the final project was to build a mobile application that is able to perform real time execution using a mobile camera as the source for the input data. After a brief overview on the given time frame for semester two I've decided to change the overall concept of building a mobile application into a web application. Web application execution will not be performed on real time and will take an input video instead.

In both the prototype and the final web application input data can be taken using either a mobile camera, digital camera or a go pro. Digital camera with the best resolution usually sell for the highest price, but nowadays mobile cameras offer excellent resolution for the object to be detected. The main rule to follow when finalizing the input video is if the basketball and athlete can't be recognized by you, then there's a high probability the system can't detect any objects.

2.2.3 Database

Convolutional Neural Networks is such a big architecture and will only continue to grow during its training phase. Implementing persistence in our application saves us processing time by re-using the trained model rather than training a new model. JSON which is the standard file exchange format will be used in both prototype and final application. NoSQL a cheap non-relational database that follows no structure is chosen as the database for the application because of its ability to scale with increasing data. This enables us to train a pre-trained model efficiently and the use of JSON as the main storing format gives us easy access to our data set.

2.2.4 YOLO Object Detection

You only look once algorithm or YOLO for short is currently the fastest object detection algorithm today. YOLO is a one-stage detector strategy that only looks at the input image once before detecting any classes on the image. It does this by separating the image into an S by S grid where each grid cell is responsible for predicting five bounding boxes. Bounding box in short is a description of the rectangle that encloses the object in the image. Lastly YOLO outputs a confidence score telling us how certain the system is that the predicted bounding box actually encloses an object. (Sharma, 2018)

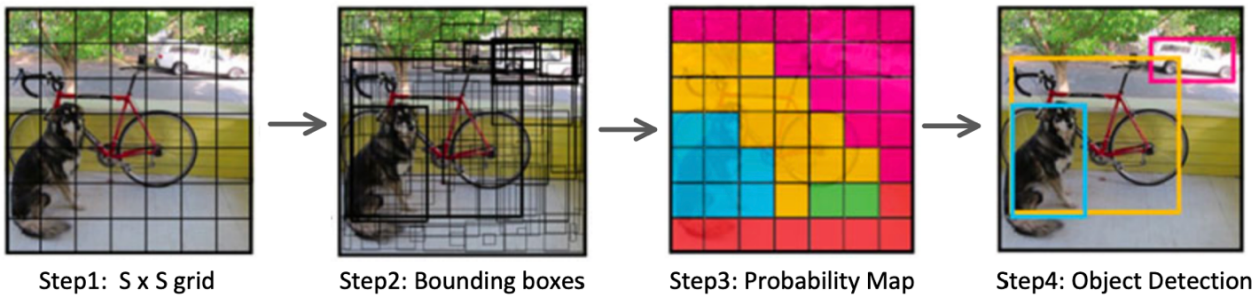

Figure 2.1: *YOLO Object Detection*

2.2.5 R-CNN Object Detection

R-CNN or Region CNN is another method for object detection in deep learning. This method can be summarized into three steps. First different image proposal is created for a single image. Secondly for every proposed image an SVM classifier is used to detect what class the object on the image belongs to. Lastly every classified image is overlapped and the proposal with the highest score it picked. R-CNN is currently the second fastest and most accurate detection method where it once scored 54% on PASCAL VOG 2010. (Ananth, 2019)

2.3 Build Design

In this section, we present the main framework that the final application follows as depicted in Figure 2.2. The detection system in mainly composed of four main parts:

1. Shot Attempt Classification: is the process of classifying whether the athlete is attempting to shoot the ball.
2. 2D Basketball Detection: is the process of detecting the basketball during a shot attempt and detecting whether the basketball goes in the basket or not.
3. 2D to 3D Conversion: perform algorithms to try convert 2D frame into 3D, this enables the user to see the location of every shot attempts.
4. Mathematical Computation: simple calculation of every shot attempts and output a statistical representation of every shots made or every shot missed.

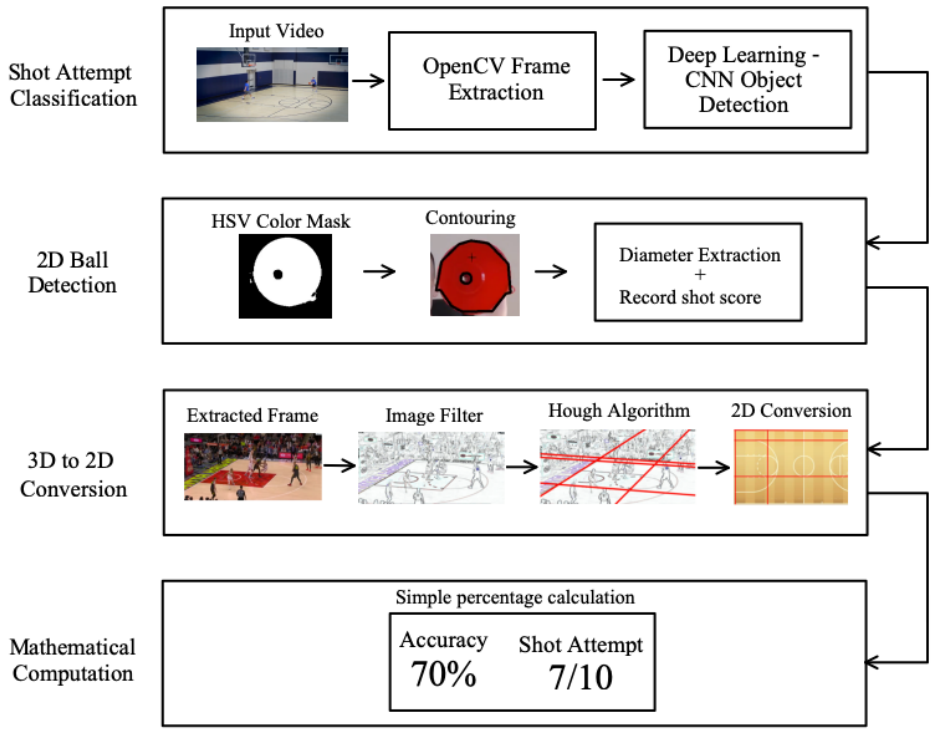

Figure 2.2: *Build Design Framework*

2.3.1 Shot Attempt Classification

Given the input video, a total of five frames per second will be extracted as image using Python's library OpenCV. Some objects might be blurred at certain times therefore extracting more frames results to better detection accuracy, the reason is simply because the more frames to process the smaller the percentage error and there is a high probability that the next frame is able to compensate for the blurred object. Extracted frame will then be save into the database as it is needed later during the 2D ball detection stage and during the 2D image conversion of the basketball court.

Every frame is then fed as input to a trained convolution neural network. The CNN classifies every frame to check whether they belong to the one and only class in the network which is the shooting class. The CNN outputs multiple classified frames and will then be stored to a different dictionary on the database. The output will be used in later stages when locating the position of the player during his or her shot attempt.

2.3.2 Basketball Detection

In this stage of the build process frames are accessed from database. The main goal during this stage is to confirm if the basketball goes in the inside hoop and through the net or if it's a missed shot. A counter method increments every time the detection system detects a shot attempt from a user, this records the total attempt which will be used on the last stage where the user's shot accuracy is calculated.

HSV Colour Map

HSV or Hue, Saturation and Value is a different approach to represent a colour and a lot easier in terms of object detection based on colour compared to RGB colour -space. Normally colour detection algorithm iterates through every pixel in the image and searches for a specific colour, but with HSV the algorithm search in terms of colour purity and colour position.

The RGB colour model uses three main colours (red, green and blue) each having a value that ranges from 0 to 255. RGB forms its colours by adding different values of red, green and blue. The HSV colour model uses three dimension, hue represents the colour, saturation represents the dominance of that colour and value represents the brightness of the colour.

Shape Contour and Diameter Extraction

Masking is the process of keeping any pixel that is in the HSV colour range and discarding every pixel outside the colour range. When masking is completed a contour method is applied to insert hard lines around the masked pixels, the algorithm works as follows:

- a check method is performed to ensure that the input image contains at least one contour.
- if a contour is found the system computes the minimum enclosing circle using the detected contour as the initial point and then computes the centre of the circle (x, y).

In this paper finalizing the method on how to figure out whether the ball went through the basketball hoop was the trickiest section to research. The method pointed out by my supervisor was the most ideal approach and decided to stick to it. The method proposed was comparing the diameter of the detected basketball to the width of the basketball hoop, whenever the ball intercepts the hoop and both widths matches then it counts as a basket otherwise it doesn't count.

2.3.3 3D to 2D Conversion

The main objective of 3D to 2D conversion is to pinpoint the location of the athlete on the basketball court. When this phase is fully working users can view every position of shot attempt and have a clear representation of which position are made or missed shots, shown in Figure 1.2.

In this stage the Hough Transform algorithm is used to detect the hard lines on the basketball court. Since dimension on a basketball court is known, we are able to map the 3D point coordinates to a 2D plane.

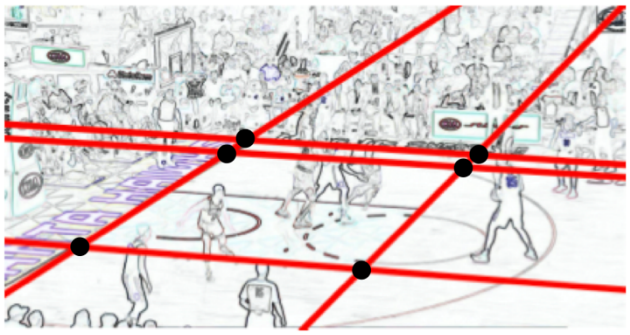

Figure 2.2: *3D Court Image*

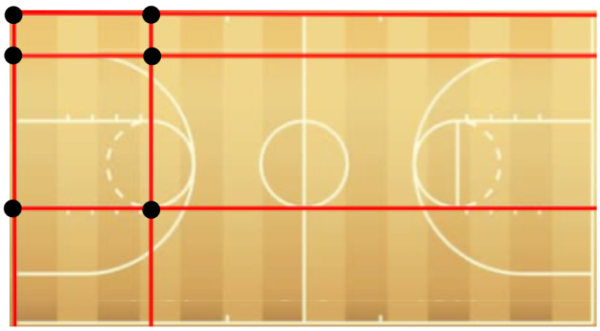

Figure 2.2: *2D Court Image*

2.3.4 Mathematical Computation

This stage computes the percentage of shot attempt of a single video using basic math calculations, calculated percentage will give users an overview of their current shooting ability.

3 Prototype

Currently the prototype is a one-page Python code that is only able to perform ball detection. The prototype can be executed through the terminal by navigating to the file directory and by using the command 'python3 prototype.py'. The prototype when executed opens the built-in video camera on your PC and starts to detect various colour of orange. In Figure 3.1 a screenshot of the prototype working is shown, during the test I used my phone to test different colour range of orange with various light exposure.

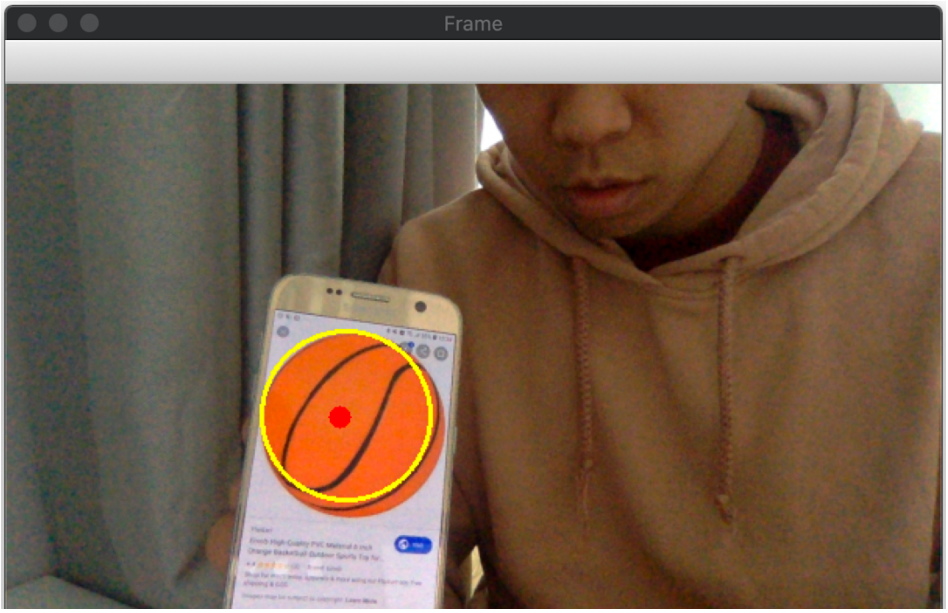

Figure 3.1: *Prototype Test*

4 Conclusion and Discussion

4.1 Limitations

The duration of given time for the development phase is only eight weeks, this given period might not be enough specially when deep learning algorithms are so advance and might take more time than planned to fully develop. Gathering of the data set alone will require one week, also another week or more is mostly needed to build and train the convolutional neural networks. If debugging was to be included it would add another whole week just to locate bugs in the architecture and to fix the architecture, meaning with the given time frame there are time limitation for each stages of the build.

Another limitation pointed out is that with MATLAB a licensed needs to be purchased before the software is ready to use. This did open up a better option which was to use Python instead on the initial software and programming language picked in the project proposal.

The given time frame limits the idea of building a mobile application, which would've been more reasonable for athletes as they can use the application anywhere when they train. Mobile application tends to be more successful compared to a web application mainly because people prefers quick and easy access to application nowadays.

4.2 Improvements and Recommendations

One main improvement that can implemented onto the application is to add a better overview of their shooting ability by using graphs. This can be done by adding a calendar function on the application that automatically set the current date on the app and whenever the app is used the data of shot attempts is stored along with the date. A weekly graph of the days versus the percentage can then be produced.

Coloured heatmaps can also be implemented on the application. This feature adds a colour label on the location of shot attempts on a two dimensional image of the court. The colour is then based on the overall percentage shots taken by far and can range from green being average and reg being hot.

4.3 Conclusion

This paper showed that deep learning is a subfield of a bigger architecture called machine learning and that the concept of computer vision can be tackled with deep learning techniques. Using simple image classification and localization objects can be detected on an image. Writing neural network algorithms to train a model is the process of training the model to recognized a group of objects or classes present on an image. During the process of detection a probability score is used to determine which class the object belongs to, then a box is inserted around the object along with its class name to classify that it is detected. This paper concluded that using Python is more reasonable compared to MATLAB because of its first it is open source, second is its flexibility as a programming language and lastly Python has a massive community for support. Python also offers multiple advance libraries which can be implemented onto our project to lessen the line of code and improve execution time.

Bibliography

Ananth, S. (2019, April 27). *R-CNN for object detection.* Retrieved from Towards Data Science: https://towardsdatascience.com/r-cnn-for-object-detection-a-technical-summary-9e7bfa8a557c

Chollet, F. (2017). *Deep Learning with Python.* Manning.

Gibson, A., & Patterson, J. (2017). *Deep Learning.* O'Reilly Media, Inc.

Goodfellow, I., Bengio, Y., & Courville, A. (2015). *Deep Learning.* London.

Gupta, N. (2019, October 8). *Why is Python Used for Machine Learning?* Retrieved from Hackernoon: https://hackernoon.com/why-python-used-for-machine-learning-u13f922ug

Josephson, A. (2018, June 11). *The Economics of Personal Training.* Retrieved from Smartasset: https://smartasset.com/career/the-economics-of-personal-training

Koph, D. (2017, October 18). *Data analytics have made the NBA unrecognizable.* Retrieved November 2019, from Quartz: https://qz.com/1104922/data-analytics-have-revolutionized-the-nba/

Press, G. (2016, December 30). *A Very Short History Of Artificial Intelligence (AI).* Retrieved from Forbes: https://www.forbes.com/sites/gilpress/2016/12/30/a-very-short-history-of-artificial-intelligence-ai/#35821d5b6fba

Sharma, P. (2018, December 6). *A Practical Guide to Object Detection using the Popular YOLO Framework – Part III (with Python codes).* Retrieved from Analytics Vidha: https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/