FINAL PROJECT REPORT

MCS13

FIT 3162: COMPUTER SCIENCE PROJECT 2

MEMBERS

Matin Raj Sundara Raj (32124260)
Suah Wen Hung (30721083)
Louis Juliano (31507794)

~ 7642 WORDS ~

BadminSmash - Visual Analysis of Badminton Shot Styles

Supervisor : Professor Raphaël Phan

Table of Contents

1.0 Introduction

In this era of digitalization, advancements in technology have revolutionized the way we analyze sports events. This is said because it provides valuable statistical data to enhance our understanding during a live game. However, such technologies are often proprietary and limited to enterprise-based solutions. This limits accessibility to small independent researchers or sports enthusiasts who want to use technology to gain statistical insights into their sports matches. Thus, the primary aim of this project is to bridge that gap by leveraging open-source systems like TrackNetV2 to develop a more widely available and low-cost approach to data collection and analysis in the context of broadcast match videos.

The ultimate goal of this project as set by the supervisor, is to build on top of TrackNetV2 and enhance its feature to detect and predict the shot styles of players. This includes the classification of forehand, backhand, net drop and serve shots. Given an input video file of a badminton match, the software should be able to annotate the predicted shot styles of the player and produce an output video file with all the annotations. This software will provide a platform for advanced analysis, enabling deeper insights into player movements, strategies, and performance. To ensure efficient project development, we have adopted the Agile framework, allowing us to break down the tasks into time-boxed Sprint iterations where the focus is on developing a working product in an incremental manner.

In this final report, we will critically discuss the project's outcomes, methodologies, software deliverables, current limitations and future plans.

<u>The Team</u>

Matin Raj – Project Manager

Responsible for the planning of resources, aiding team members to overcome difficulties faced and ensuring the deliverables are completed by the deadlines set. In line with the Scrum framework adopted, he is also assigned the role of Scrum Master who ensures that the team adheres to the project development methodology and facilitates Scrum ceremonies.

Louis Juliano – Technical Lead

Responsible for providing an architectural and design direction for the code base of the project. In line with the Scrum framework adopted, he is also assigned the role of product owner who is responsible for managing the product and sprint backlogs while liaising with the supervisor.

Suah Wen Hung – Quality Assurance

Responsible for the product’s performance, accuracy and overall functionality. Performs rigorous testing after each feature implementation to ensure the product delivered is up to standards.

2.0 Project background

2.1 Overview

Played since the 19th century, badminton is highly regarded as one of the most popular racquet-based sports in the world. The nature of the sport stimulates extreme competitiveness from players across the globe with Gangal and Raje (2014) highlighting the current need for games to be as fair as possible in order to produce a meaningful and justifiable outcome. This is where the rise of modern technologies, greatly benefited the sport by providing umpires with various real-time analyses of ongoing aspects of the match such as the trajectory and location of the shuttlecock with high precision. For example, With the birth of the HawkEye technology, Baodong (2014) explained that high-end cameras have been installed in stadiums to capture footage of ongoing matches where the data collected are processed in real-time to be shown as instant replays for the justification of the refereeing decision to ease a player's challenge as well as the supporters watching live at home.

The usage of such technologies stems beyond competitive matches and is recognised to be helpful for the development of an athlete, as well as for match analysis. Yunwei and Shiwei (2019) justified that video analysis technology used in sports could well provide trainers with more effective training methods in order to coach and improve their players. However, it is noted that access to such state-of-the-art technologies is often limited due to cost barriers as most are proprietary software tools and others need expensive sensor equipment to be set up. These technologies are inaccessible to small independent researchers who would want to perform research in the field of badminton and to enthusiasts who want statistical data for their non-competitive badminton matches.

With severe feature limitations on existing open-source badminton analysis software, this fuels our motivation to build on top of such open-source platforms like TrackNetV2 by enhancing its capabilities to perform much-advanced badminton analysis. TrackNetV2 provides a general-purpose sports ball tracking system. Thus, it is prudent to focus on the badminton players to extract statistical information that can be used as means to improve their gameplay. The analysis of shot types in a badminton match is crucial in informing the player of the opponent's playing style which may help them in formalizing a defence and attack strategy. As such, the team was motivated to detect and predict different shot types such as serve, forehand, backhand and net drop shots, enabled by computer vision and machine learning.

2.2 Literature Review

The literature review has been updated to support the implementation of our software and what the team has produced. Discussions from new research papers are highlighted in blue.

In recent times, there has been an increasing trend to gather data and extract game statistics which could act as a tool for those interested in the sport. It is highly agreed within the sporting community that the provision of real-time data analysis could well take the immersion of the game for fans and enthusiasts to the next level. Chu and Situmeang (2017) stated that game statistics could peak the point of interest in an ongoing match. For example, broadcast matches could show the rally of a point, the speed of the shuttlecock, or even the average position of a player. There has also been an idea proposed to utilise augmented reality as a technique to overlay virtual objects onto an ongoing broadcast match (Mahmood et al., 2014). It was believed that this could engage viewers right into the action of the game by displaying simple data or game scores right onto the court.

While one benefit of showcasing game statistics comes in the form of entertainment, a much more meaningful application is possible with the advent of sports analysis tools. This is where the use case for our product shines. In their research paper, Su and Liu (2018) stated that tactical analysis is an important factor in identifying the style of an opponent's tactics which allows the improvement in the performance of a team or an individual athlete. With the usage of technology, Rahmad et al. (2018) mentioned that the analysing process of a previously played match may dictate coaches into managing the player's position at any point of a game by fixing some prone errors which may be discovered through the analysis.

Apart from that, analysing direct badminton videos can also improve the teaching methods of physical education teachers at various school levels. According to Xipeng et al. (2022), the traditional teacher-led teaching methods of explaining the rule of the sport, understanding simple scoring techniques and practical teacher-to-student training would enforce students to only learn through idealising their understanding of badminton action and naturally having to rely on the teacher's guidance without getting the point themselves. In this case, incorporating technology during the teacher's teachings should eliminate some of the negative aspects of traditional methods. For instance, Lin et al. (2021) stated that visualisation of abstract sports concepts could be achieved through using technology where teachers may be assisted in teaching students the accurate postures and strengths to be used in any situation.

However, despite the rising research done in match video analysis, it has been noted that there is still a low impact in the study of Badminton videos specifically. Tan et al. (2016) stated that further scientific research is more prominent in other racquet sports like tennis despite badminton being the fastest racquet sport with regards to the shuttlecock speed. Chu and Situmeang (2017) further iterate that a lot of the past research done has been generalised under racquet sports, however, it should not be the case as badminton is played under a different playing pattern where the speed of the game differs from other racquet sports. Even the first released version of TrackNet, placed its focus on generalising tracking tiny objects moving at high velocity under various sports applications, starting with tennis matches (Huang et al., 2019). Thus, this has been one of the team's motivations in pursuing the development of a software analysis tool catered specially for the sport of badminton.

A previous attempt in performing badminton match analysis has been done by Chen and Wang (2007) where they uncover a 2D analysis system with the help of a camera to capture badminton footage from a high diagonal viewing point to cover the entire playing field. This setup actually came back particularly accurate with the collection of barycentre images which achieved an almost 100% detection rate of the shuttlecock trajectory. Under that shot detection method, the input taken are usually frames of a badminton match from a fixed camera which would be fine as long as the player and shuttlecock stay within the frame while the entire analysis is performed. However, when that condition is no longer met, some frames could return inaccurate results as the system mostly would not cater for such situations in their current rendition. As such, Yoshikawa et al. (2021) proposed a new system to instead use the frames of players' skeletal information as they navigate around the court in order to estimate the position of the shuttlecock, even when out of frames. The system mostly only managed to assess simple shot detection through the connection of player position in conjunction with the estimation of shuttlecock hit timing.

Where our product differs is, our analysis is focused on the player itself. This is because as we built our system on top of TrackNetV2 which already tracks shuttlecock and its trajectory, we are much more incentivised in extracting player-related statistics which can provide a much bigger impact. In this regard, Liao et al. (2022) acknowledge that pre-match information collection is crucial to understanding the opponent. This research method enabled by technology directs coaches and researchers to find the regulations of opponents' technique skills, offence and defence tendencies, and how they win or lose points. For example, the highlighting feature of the software that we have developed is the shot style classification of players. Running our software on a badminton match video of the opponent will allow interested parties to gain insights on which shot types the opponent struggles with thus losing a point, and which shot types are their strengths, thus winning a point. Utilizing this information, coaches and researchers could develop a proper training plan and efficient match strategy for players.

Apart from formulating strategies to combat opponents during a badminton match, analysis tools enabled by machine learning can also help stimulate an organic learning process for the players involved. In the field of badminton, measuring how good the shot is in a rally is important for decision-making and tactic investigation. Especially focusing on the last few shots, utilizing statistical data to investigate performance such as forced or unforced errors for analyzing the behaviour of a player can indicate how likely it is that they can score or lose a point Wang et al. (2021). Mistakes or unforced errors are inevitable but through training, they can be reduced from occurring with high frequency during competitions. A key contributing factor to unforced errors is the player's technique and stance. Strengthening the practice of basic techniques, the proficiency, agility, balance, speed, and strength of the handling skills should be prioritized when trying to overcome unforced errors Liao et al. (2022). In this context, our badminton analysis software annotates a skeleton on the player's body through the usage of the MediaPipe library. The balance and stance of the player when performing a particular shot that leads to an unforced error such as serving, can be inferred from the skeleton landmarks which visually illustrate the position and angle between joints to the user. This information can be used by a trainer to correct the player's technique in performing a particular shot.

It is well-known in the badminton community that the outstanding performance of athletes depends on not their skill and training but also the mastery of the opponents' superiority, inferiority, and even tactics and emotion in the games. Videos can contain a large amount of information and traditionally, tactical data were manually labelled from video, but this approach is time-consuming and prone to errors, Hsu et al. (2019). A research was done by Jin Qiu in 2022 demonstrating the characteristics and advantages a player would have if shots were performed effectively. They stated that a successful serve can gain the initiative of a match and put pressure on the opponent. At the same time, an effective serve can force the opponent to use the technique expected by the server to return the shuttle. However, the methodology used to gather the information to conduct this research was through manual slow playback and pausing frame by frame to record the type of technique and the landing point. They acknowledge that an automated tool would be much more accurate and efficient. This justifies our aim in catering our software towards small independent researchers. Besides, this research shows that our classification of the serve shot style is very crucial in informing the user of their technique which has the potential of gaining an initial advantage.

Our product is aligned with an advanced research done by Chu and Situmeang (2017) where they developed some important components in performing stroke classification depending on a player's posture and subsequently a strategy classification to determine the tactic a player is utilising in an attempt to defeat their opponent. However, during their testing, it seems to only achieve an average accuracy in the range of 80%, which the duo concluded could be further improved in the future with a strategy modelling which is more elegant. Still, the studies specifically in the provision of match analysis seem to be somewhat lacking, with more research seemingly leaning towards building on player detection or playing robots instead. Thus our product has introduced a relatively novel analysation tool for the sport of badminton where the information extraction is based on the classification of player shot styles.

All in all, this literature review discussed the various possibilities of bringing technology to perform analysis of badminton matches with the gaps that exist in previous implementations and how the team has managed to bridge this gap through our software. The team reiterates that bringing such technology to easy public access for researchers and enthusiasts alike is crucial in stimulating badminton skills for research or professional purposes. At its current stage, while there has not been much badminton-specific research in video analysis, the studies aforementioned shed some light on the analysis of shuttlecock tracking, pose detection and even prediction in out-of frames tracking. Furthermore, there has also been a few undiscussed studies due to being out of the scope of analysis which instead looked at more physical and practical usage such as putting on wearable tech for stroke detection or even a functioning player robot which can play a full-length match against a human opponent. Thus the development of our software has proven to be important in providing low-cost options to small researchers and badminton enthusiasts alike to obtain badminton match statistics built only using computer vision and machine learning. As such, the badminton match analysis tool created by the team will eventually be published in the wild to help the studies of others.

3.0 Outcomes

3.1 What Has Been Implemented

Based on our initial requirements, we have implemented almost all of the proposed items with one minor feature substitution. The table below summarizes all the software requirement implementations.

Functional Requirements	Is Implemented	Reason
The product must be able to correctly predict the player's backhand shot pose	YES	-
The product must be able to correctly predict the player's forehand shot pose	YES	-
The product must be able to correctly predict the player's net drop shot pose	YES	-
The product must be able to correctly predict the player's smash shot pose	NO	We realized that the smash pose is too similar to the forehand pose which confuses our prediction model. As a result, we compensate for this by classifying a new pose called Serve
The product must be able to utilise TrackNetV2 to track the ball	YES	-
Collect enough data to train the machine-learning model	YES	-
Non-Functional Requirements	**Is Implemented**	**Reason**
The product must be able to detect the player's position in real-time	YES	-
The product must be able to show the player's skeletal pose in real-time	YES	-
The product must use a well-trained model to accurately predict badminton shots to display match statistics	YES	-

Additional Implementations during development	Is Implemented	Reason
The product must have a UI for ease of use	YES	We added this because we figured that it would be easier for user to use a user interface instead of running the product with terminal
The product must be able to correctly predict the player's serve shot pose	YES	To compensate for previous smash shot pose
The product must be able to predict both left and right handed player	YES	To ensure predicted shot styles are accurate regardless of which hand the player holds the racquet

Table 1: Requirements Table

3.2 Results Achieved and Related Discussion

We have successfully developed a functional machine-learning model that is capable of predicting skeletal poses. This model has undergone rigorous training and testing to ensure its accuracy and reliability. Additionally, we have implemented a code that utilizes the powerful capabilities of MediaPipe to estimate the player's pose. By combining these two components, we were able to accurately predict and analyze skeletal poses in real-time. This project enables us to effectively track and analyze the player's movements, thereby enhancing the accuracy and precision of our shot style predictions.

MediaPipe Pose Estimation

MediaPipe serves as a crucial tool in our project, primarily utilized for predicting the skeletal poses of players. By utilizing the capabilities of MediaPipe, we were able to extract skeletal poses from the captured video, with each joint referred to as a landmark. These landmarks provide essential information about the player's body position and movement.

To predict skeletal poses, we have developed a machine learning model. The construction and specifics of this model will be discussed in the upcoming section. This machine learning model, combined with the data obtained from MediaPipe, enables us to accurately predict and analyze the skeletal poses of players in real-time, forming the foundation for subsequent shot style predictions and analysis.

Model

We have successfully developed a machine learning model that excels in predicting four distinct shot styles: Forehand, Backhand, Net drop, and Serve. To achieve this, we applied the Random Forest Classifier (RFC) algorithm, which has demonstrated remarkable performance throughout our development process. The Random Forest Classifier exhibits an impressive accuracy range of 70% to 76%, with an average accuracy of 74%. This level of accuracy provides a solid foundation for reliably classifying and identifying the different shot styles executed by players. By leveraging the capabilities of the Random Forest Classifier, we can confidently analyze and categorize shots, enhancing the overall accuracy and effectiveness of our project.

The table below shows the confusion matrix for the RFC algorithm used. Highlighted in green is the True Positive and True Negative of the model. False positives and negatives are highlighted in red.

	backhand	forehand	netdrop	serve
backhand	5	4	3	0
forehand	0	45	5	1
netdrop	4	11	21	0
serve	0	2	2	16

Table 2: TP/FN table for RFC Algorithm

3.3 How Are Requirements Met

Functional Requirements	How are they met?
The product must be able to correctly predict the player's backhand shot pose	We have successfully constructed a model with an accuracy of 74% to predict backhand shots
The product must be able to correctly predict the player's forehand shot pose	We have successfully constructed a model with an accuracy of 74% to predict forehand shots
The product must be able to correctly predict the player's net drop shot pose	We have successfully constructed a model with an accuracy of 74% to predict drop shot shots
The product must be able to utilise TrackNetV2 to track the ball	TrackNetV2 is run before our model. We have integrated both codes together
Collect enough data to train the machine-learning model	We gathered frames from official BWF match videos and sorted them manually. Our trained model has an accuracy of 74%. This high

	accuracy means we had enough data to train our model.
Non-Functional Requirements	How are they met?
The product must be able to detect the player's position in real-time	The program shows the prediction of the badminton shots while playing the video
The product must be able to show the player's skeletal pose in real-time	MediaPipe annotates the skeletal pose of the player
The product must use a well-trained model to accurately predict badminton shots to display match statistics	Our trained model has an accuracy of 74%. This high accuracy means are able to predict poses accurately
Additional Implementations during development	How are they met?
The product must have a UI for ease of use	Use of TKinter library
The product must be able to correctly predict the player's serve shot pose	We have successfully constructed a model with an accuracy of 74% to predict serve shots
The product must be able to predict both left and right handed player	We have added a new functionality that allows user to choose whether they are predicting left handed or right handed player. This is enabled through a checkbox in the UI to indicate to software if player Is left handed.

Table 3: How each requirement was met

3.4 Justifications of Decisions Made

Algorithm Changes

After careful evaluation and testing of various machine learning models, we have made the decision to utilize the Random Forest Classification algorithm instead of the K-Nearest Neighbors Algorithm (KNN) as proposed in our initial project design. Our selection was based on the performance of the algorithms. Random Forest Classifier has demonstrated that they have the highest accuracy among all the models tested.

The Random Forest Classifier consistently achieved an impressive accuracy ranging from 70% to 76%, with an average accuracy of 74%. This high level of accuracy allows us to have confidence in the model's ability to accurately classify shot styles. Moreover, the algorithm performs exceptionally well in real-time scenarios, further solidifying its suitability for our project. By employing the Random Forest Classification algorithm, we can confidently classify and predict shot styles with high accuracy, thereby enhancing the overall effectiveness and reliability of our system.

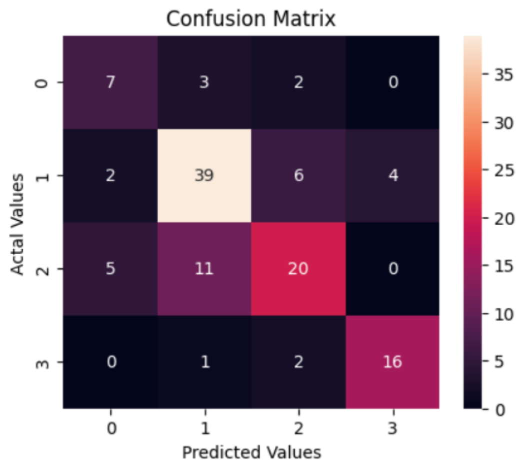
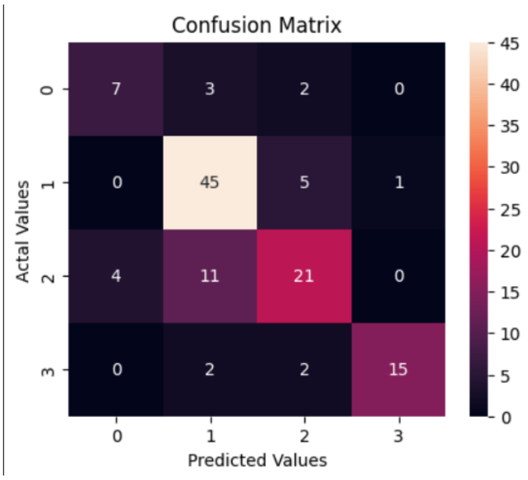
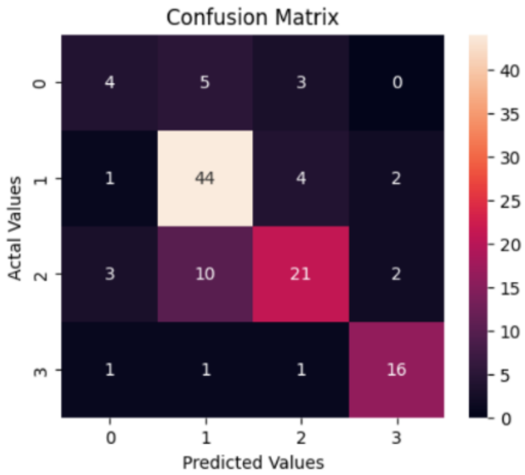
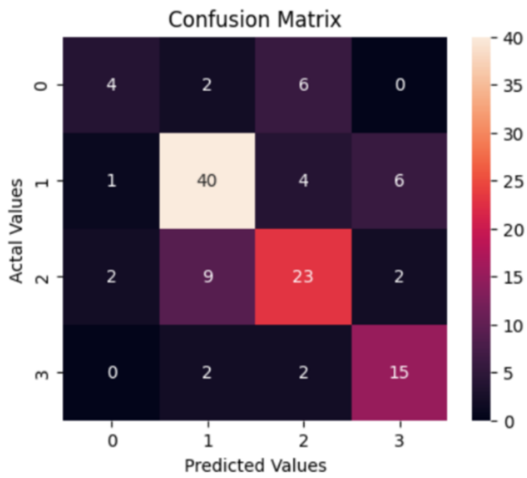
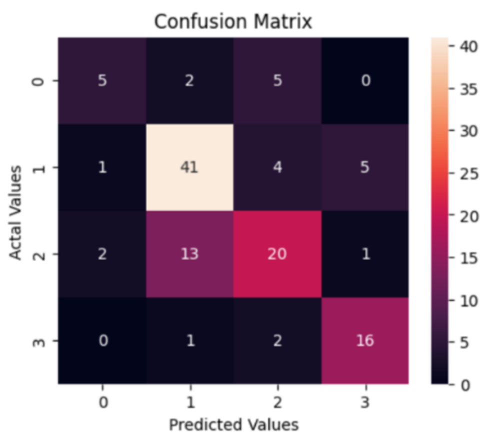
<table>
<tr><td>K-Nearest Neighbors TP/FN Table</td><td>Random Forest Classifier Algorithm</td></tr>
<tr><td>Accuracy: 69%</td><td>Accuracy: 74%</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Gradient Boosting Classifier</td><td>Ridge Classifier</td></tr>
<tr><td>Accuracy: 70%</td><td>Accuracy: 69%</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Logistic Regression</td></tr>
<tr><td colspan="2">Accuracy: 69%

</td></tr>
</table>

Table 4: Confusion Matrix Comparison Between Algorithms

Requirements Changes

The first requirement change we made was to substitute the prediction of smash styles to serve styles. We recognized that the forehand style and the smash style have a significant resemblance in terms of player pose, making them difficult for our model to distinguish accurately. Considering that the smash is a subset of the forehand style, we made the decision to classify smash shots as forehand shots. To compensate for this adjustment, we introduced a new shot style into our system: the serve shot style. By incorporating the serve shot style, we aim to enhance the accuracy of our shot style predictions. This addition allows us to differentiate between the distinct serve and forehand shot styles, providing more comprehensive insights into player performance.

One of the significant requirement changes we have implemented is the addition of an Input Graphics User Interface (GUI) element to our product. Previously, TrackNetV2 operated solely through the console without any graphical interface. However, based on our experience and user feedback, we recognized the need for a UI element to enhance the usability and ease of interaction for end users.

By implementing an Input GUI, we aim to provide a more user-friendly experience. Users will have visual elements, controls, and interactive features that facilitate easier navigation, input, and interpretation of the results. This GUI will serve as a medium for users to interact with the product seamlessly, empowering them to utilize our product more efficiently and effectively.

3.5 Limitations of Project Outcomes & Possible Improvements for Future

One of the primary limitations of our project is the significant slowdown in performance. This issue arises from the current implementation of TrackNetV2, which requires running the input video twice. Additionally, our product itself further contributes to this slowdown by processing the video for a third time. Running the video multiple times places a heavy computational time on the system, leading to decreased performance and slower execution times. This can result in delays in processing and analyzing videos, negatively impacting the user experience and overall efficiency of the project.

Addressing this limitation is crucial for optimizing our project's performance. Exploring alternative approaches that reduce the need for redundant video processing or optimizing the existing algorithms can potentially mitigate the slowdown. By finding more efficient ways to process and analyze the input videos, we can improve the overall speed and responsiveness of our project, enhancing the user experience and increasing productivity.

To address this limitation, we have devised a plan to enhance the performance of our project. In the future, we aim to merge the processes of TrackNetV2 and our product into a single loop. By doing so, we can eliminate the need for running the video three times and instead, run the video only once. This approach will significantly reduce the computational time and enhance the overall efficiency of our project. By integrating all the processes into a single loop, we can optimize the execution flow, eliminate redundant calculations, and minimize processing time. This optimization will result in faster performance.

Another limitation of our project is the lower accuracy in predicting the backhand and serve shot styles compared to the forehand and net drop styles. This limitation arises from the insufficient amount of data available for training and evaluating the model specifically for backhand and serve shot styles. To address this issue and improve the accuracy of our predictions for backhand and serve shot styles, we plan to collect more data specifically focused on these styles. By gathering a larger and more diverse dataset that includes backhand and serve shots, we can enhance the model's ability to learn effectively for these particular styles.

4.0 Methodology

4.1 Design

The project's approach has undergone a slight change compared to the initial proposal. In the previous proposal, the plan was to obtain a predicted pose from the pose estimation function. However, during the course of development, it became evident that this approach was not feasible. Consequently, we have modified our approach to adopt a similar alternative. The revised approach now involves leveraging MediaPipe, a framework used for predicting skeletal poses. The skeletal pose obtained from MediaPipe serves as input for our trained model, which then predicts the corresponding shot styles. This new approach ensures the integration of MediaPipe and our trained model, enabling us to accurately classify and determine shot styles based on the skeletal poses.

After extensive development and experimentation, we decided to change the model algorithm used for training. The previously proposed K-Nearest Neighbours (KNN) algorithm has been replaced with the Random Forest Classifier. Through our iterative process, we discovered that the Random Forest Classifier has been performing the best among other models in terms of accuracy. It has yielded impressive results, with accuracy ranging from 70% to 76%, and an average accuracy of 74%. This improvement in accuracy reinforces our confidence in the Random Forest Classifier as the optimal choice for our project, ensuring reliable and precise predictions of shot styles based on the given input data. Furthermore, our approach to data collection and gathering was shifted from relying on readily available datasets found online to manual cropping of frames from official BWF match videos and sorting them into different shot styles. The table below summarizes the final design flow of our prediction model development.

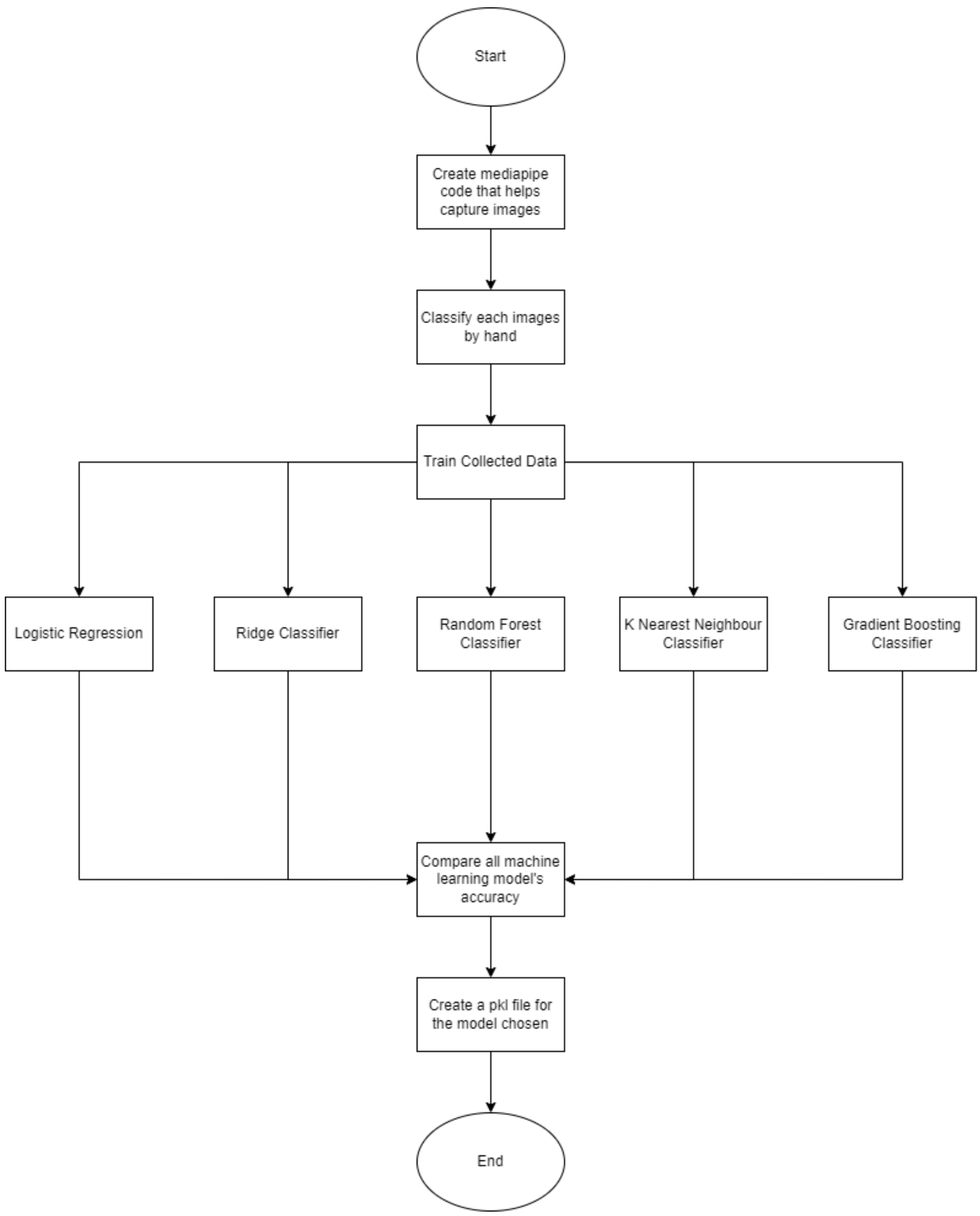


Diagram 1: Flow chart of Final Software Design

4.2 Software Tools and Development

i. **Python**

The team used Python programming language as it is widely used in the fields of AI and Machine Learning and is known for its simple structure and syntax, making it highly readable for large projects. Moreover, Python offers support for the necessary libraries used in machine learning and tracking, which are crucial for player tracking and classification models. Additionally, since the open-source code we worked with, TrackNetV2, was written in Python, it makes Python the logical choice for our team to enhance the badminton shot detection software.

ii. **Visual Studio Code**

Among all team members, Visual Studio Code is the widely used integrated development environment (IDE) due to its various extensions that streamline the coding process. One notable feature of Visual Studio Code is its Live Sharing functionality, allowing multiple users to collaborate on the same code file simultaneously. This feature is particularly advantageous for remote project development, providing significant benefits to the team during live code-sharing sessions.

iii. **GitLab**

GitLab, an open-source version control system, is widely recognized and understood by all team members. The team frequently push and pull changes to the repository whenever they complete their tasks remotely so that other members are in sync with the development of the project.

iv. **OpenCV**

OpenCV is an open source library for computer vision and machine learning. It proved to be a valuable asset for the team as it aids in the manipulation and processing of both images and videos, serving the requirements of the project effectively. Furthermore, OpenCV plays a crucial role in TrackNetV2, where it is used for video capturing in for its ball shot detection systems. Its integration with the project enhances its capabilities and ensures a comprehensive solution for the team's objectives.

v. **MediaPipe**

MediaPipe is a crucial library that has greatly contributed to our project by providing a reliable model for estimating player's pose skeletons. By utilizing MediaPipe, we are able to extract vital information from the player's pose, enabling us to gain comprehensive insights into their body position and movements. This library's capabilities allow us to accurately estimate and track skeletal poses in real-time, enhancing our ability to analyze and understand the player's posture and movements with precision and accuracy. Additionally, this library has also assisted the project in gathering data.

vi. **Pickle**

Pickle is a widely known Python library that assists the serialization and deserialization of Python objects, enabling their conversion to and from binary data streams. This library plays a crucial role in our project by assisting us in converting our model into a binary file representation. This conversion allows us to conveniently transport and manage the model, ensuring control over its versions. With this approach, the model can be seamlessly integrated into various files or code, providing flexibility and ease of use.

vii. **Numpy & Pandas**

Both libraries, NumPy and Pandas, play essential roles in aiding with numbers and data manipulation in our project. Pandas specifically has been a great help in the creation and management of data frames, which are extensively utilized by the model to predict outcomes. With the help of Pandas, the team efficiently organized and structured the data, enabling integration with the model for accurate predictions.

viii. **Scikit-Learn**

Sklearn, also known as Scikit-learn, stands out as a highly valuable and reliable Python library for machine learning. It offers a set of efficient and valuable tools for tasks such as classification, regression, clustering, and dimensionality reduction. In our project, Sklearn has played a crucial role by enabling us to create models, split data into testing and training sets, evaluate the accuracy, apply classification algorithms, and make predictions. The utilization of this robust library has greatly contributed to the success and effectiveness of our project.

ix. **Matplotlib**

Matplotlib is a powerful library that provides users with the necessary tools to visualize and plot data frames as informative graphs. Within our team, we make use of this library to create visually appealing graphs specifically for representing the True Negative and False Positive data. By leveraging the capabilities of Matplotlib, we are able to effectively showcase these important metrics in a clear and understandable manner. The library's rich features and customization options allow us to enhance the visual representation of our data, facilitating better analysis and insights. Ultimately, Matplotlib proves to be an important library in our project, enabling us to present complex information in a visually engaging manner.

x. **Seaborn**

Seaborn, a Python library specifically designed for creating statistical graphics, is one of the impactful tools for our team. Its primary application within our project lies in visualizing the True Positive and False Negative table, enabling us to gain valuable insights from the data. By leveraging Seaborn, we are able to generate visually appealing and informative visualizations, particularly by utilizing heatmaps. These heatmaps play a crucial role in enhancing the clarity and ease of interpretation of the graph, making it simpler for us to analyze and understand the underlying patterns and trends. The rich functionalities and customization options provided by Seaborn empower us to present our data in a visually compelling manner, thereby facilitating better decision-making and enhancing the overall effectiveness of our project.

xi. **Tkinter**

The Tkinter library has played a significant role in our project by enabling the creation of a user-friendly graphical user interface (GUI). This library has proved to be crucial in designing and implementing a user interface that allows users to easily interact with our product. The library's comprehensive set of tools and widgets provides us with the flexibility to create an interface that aligns with our project's requirements and user preferences. For example, we used this library to create input and output browse buttons, a textbox to display information, and a checkbox to tick for left-handed players. These inputs would then be translated by our code and adjust its the variables accordingly. Ultimately, Tkinter empowered us to deliver a user interface that enhances accessibility and usability, making our project more accessible and enjoyable for users.

xii. **Tensorflow & Keras**

TensorFlow and Keras are two prominent machine learning libraries that offer a numerous number of machine learning models and tools, significantly benefiting our project. These libraries, especially integral to TrackNetV2, which we are seamlessly integrating into our project, play a crucial role in enhancing its capabilities. With TensorFlow and Keras, we have access to a wide range of pre-built machine learning models and tools, enabling us to efficiently train, evaluate, and deploy models for various tasks. This empowers us to build and fine-tune powerful machine learning models that can accurately analyze and predict outcomes in our project. A result of such application would be the shuttlecock ball and trajectory tracking done by TrackNetV2.

5.0 Software Deliverable

5.1 Summary of Software Deliverables

Ultimately, the main item that was delivered is a program and source code that will take the user's input of a badminton video and output the same video with added annotations. Below are the deliverables within this program in the order of its workflow: -

i. **GUI Interface**. Users will be greeted with a simple interface made using the Tkinter library to enhance user experience where users can use the path browser to select their video as input and folder as output path.

ii. **Shuttlecock detection and trajectory model**. Known as TrackNetv2, it is the base code of the program that takes the input video and outputs a video with the shuttlecock being drawn as a red dot. The team was expected to build on top of this open-source code

iii. **Player pose detection model**. This is a pre-trained model adapted from the MediaPipe library, capable of tracking a singular badminton player on the court that is closest to the camera and providing 33 vital landmark points for visual and calculation purposes.

iv. **Shot styles prediction model**. This is a Random Forest Classifier made with a custom build dataset consisting of badminton match images when players are doing a particular type of shot. The current model has the ability to predict net drop, serve, forehand and backhand.

v. **Left-handed player prediction**. User has the ability to select whether the player in the video will be left-handed where the program runs and outputs everything exactly as it normally would. The default prediction is done with assumption that player is right-handed

vi. **Badminton output video**. As the program runs, a preview screen will appear to show the real-time prediction where at the end, the video will always be outputted to the user’s intended folder path.

The final flow of our badminton shot styles analysis software is as follows:

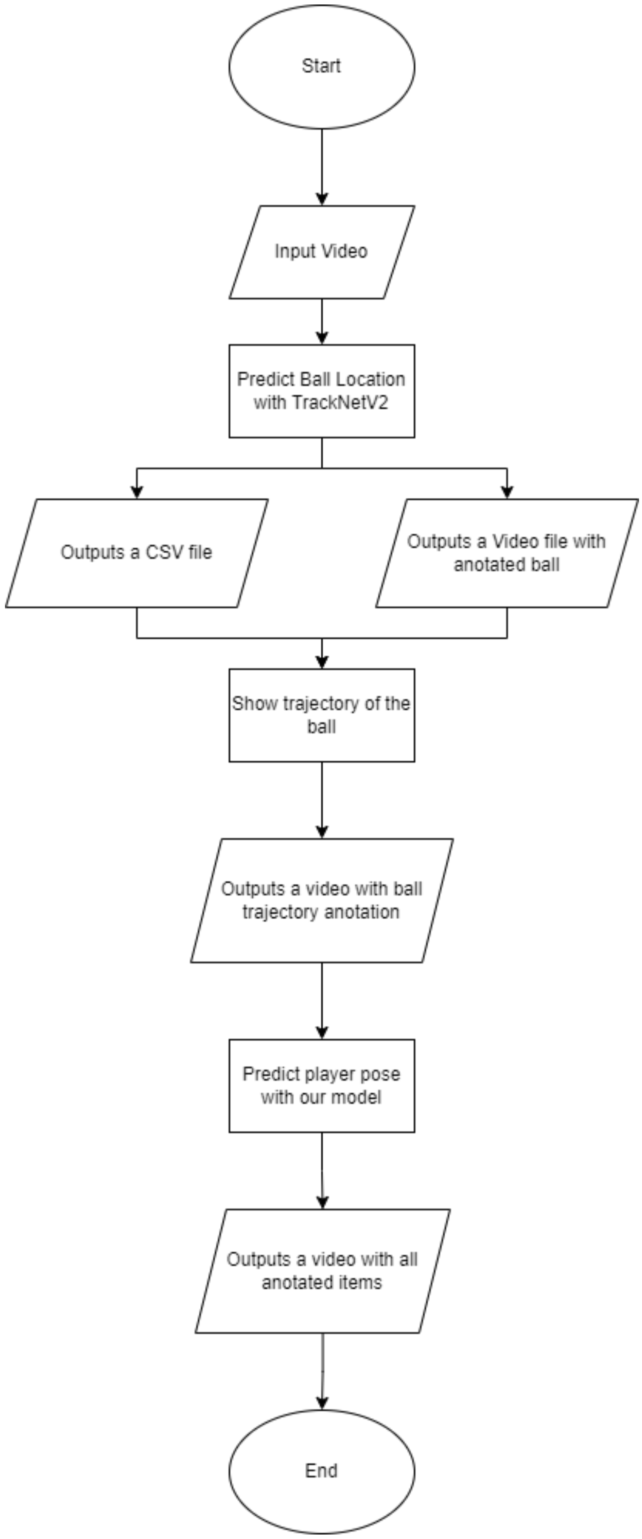


Diagram 2: Flow Chart of How the Final Product Operates

Description of Usage

A more comprehensive and detailed manual is available in the User Guide.

1. Run the program through the terminal using appropriate commands.

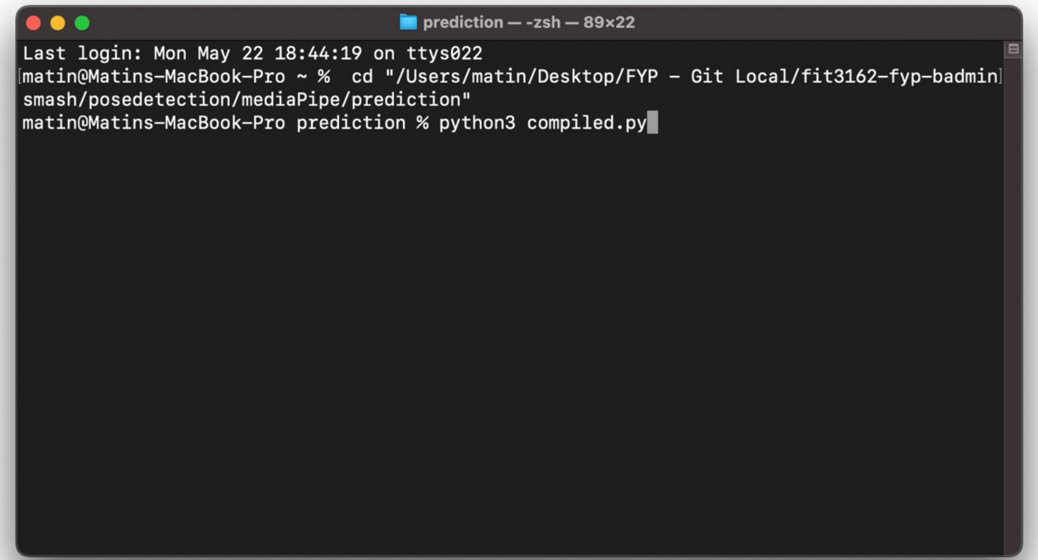


Diagram 3: Commands to run program

2. Users can use the input interface that appears to choose their input video path, output folder path and inclusion of the player's handedness, before clicking on the "Begin Analysis" button to advance.

Diagram 4: Input GUI

3. A preview screen showing the real-time prediction will appear where the user can either wait for the video to end or press ‘q’ to terminate the analysis and the video will be outputted regardless.

Diagram 5: Preview Window

4. Simply find the output video containing all the shot styles annotations and shuttlecock tracking in the output path selected.

5.2 Discussion of Software Qualities

5.2.1 Robustness

The user interface covers almost all unexpected input errors from the user. As selecting the path uses a path browser, the user is only able to choose a valid video file for the input and a valid directory path for the output. Apart from that, there is also a restriction on the input path browser to only allow the .mp4 format so the program will never need to read from other video formats. If at all the program were to encounter an error while running, error code messages will be printed onto the terminal. During the prediction process, the user's premature termination is also handled as the output video will still be added to the output path, up to the point that the process was interrupted with no issue.

The only potential issue is that users could use a badminton video with a right-handed player while ticking the left-handed player checkbox. This is not handled by the program, however, it would be obvious that the prediction will be incorrect even with no error shown, so the user should be able to realise this and start over normally.

5.2.2 Security

It was never considered throughout the project duration as users will never be at risk of any cyber danger. The program does not need any of the user's personal info, nor does it send any information online that could be a threat to the user's computer. As such, the software is secure since the program only runs locally, needing nothing else but a badminton video and an output path to function.

5.2.3 Usability

To run the program, users would need some technical ability to install the needed libraries and run Python. The provided user guide uses simple English and numbering to guide users through every step needed to get the program started.

The inclusion of a GUI (graphical user interface) made interaction with the program an easy feat for the user. There is not much information or instructions needed to use the interface, and the path browser made it simpler to select the video input and output path. Furthermore, the buttons are also relatively big and provide clear texts that signify its use like starting the analysis or exiting the program. There is also a text box that explains the purpose of the program to help the user understand.

During the prediction phase, the real-time prediction is shown to the user as an indication that the program is indeed running and predicting where they are told beforehand that terminating this process is as easy as pressing ‘q’ on their keyboard. A minor inconvenience as noted in the test report is the average wait time needed to complete the entire process as the video runs in real-time at least twice so users will need to let the program run uninterrupted for this duration.

5.2.4 Scalability

Since the project’s entire purpose was to break away from the expensive model used by official badminton matches and make it affordable within a computer machine, scalability was not considered to make this program usable within the means of real-time camera work in badminton matches. However, within the program, adding a new shot style to predict is relatively easy as all it needs is a new dataset containing images of this particular new shot style, and rebuilding of a classier able to predict everything as before as well as this shot style and running the program normally.

Attributes like player landmarks, shot style detected or ball coordinates can actually easily be utilised by storing them in a variable within the program and provide useful information like the speed of the ball, shot style counter and more. Furthermore, adding a new prediction model would be doable but it will make it slower to run as the program in its current state is already not the most time-efficient.

5.2.5 Documentation and Maintainability

The provided user guide and inline comments provided should make it straightforward for future programmers to understand the purpose and pick up the code as it is for further enhancements.

The program only ever runs within a singular Python file without the need for many functions, instead, focusing more on using various models to run its predictions and only coming up to about 300 lines of code, so any future work to improve code efficiency or change its methods should also not be much of a problem.

5.3 Sample source code

Refer to the appendix for sample source code

6.0 Critical Discussion on Software Project

Overall, the project outcome was deemed a success as it stuck true to the project proposal that was put out by the team. When compared to the initial proposal, the team did not deviate too much from the original intended implementation. Deliverables that were needed to be presented such as the pose style classifier and skeletal detection are working, and tools proposed like MediaPipe and TrackNet are being implemented within the program. Generally, any decision made which will be discussed below that changed from the initial proposal only happened due to the team either having more research done throughout the semester or realising any potential issue that needed such changes to reach the project's completion.

One hurdle discussed throughout the semester was the lack of expected support from our supervisor. This could have been due to many reasons such as inability to contact, miscommunication and mismatched timing, but the team had simply used the information provided by our supervisor during consultation sessions previously. As can be seen, the team's initial idea of having guidance and feedback from the supervisor throughout the project development was not feasible. Thus the team switched to a more independent research and development style while updating our supervisor on our progress. Secondly, the shot style classifier was originally supposed to be made with the K-Nearest Neighbours algorithm as the MediaPipe model was also built with this algorithm. During the model's implementation, various types of classifiers were tested by the team including classifiers like Random Forest Classifier, Support Vector Machine and more. However, the results showed a clear direction to using the Random Forest classifier instead due to its high accuracy in predicting badminton player pose styles as compared to the rest.

Apart from that, the initially proposed shot styles that were to be predicted were backhand, forehand, net drop and smash. While collecting data for the classifier, the team recognised that forehand and smash shared many similarities in the way that they were performed by the player, with the difference being the inclusion of jumping. As the project did not implement any player positional technology, it was difficult to train the classifier to differentiate between a forehand style and a smash style. This led to our model becoming confused between smashes and forehand shots resulting in inaccurate classification. As such, the smash style had to be combined with the forehand style where the classifier would identify both as forehand. Ultimately, this reduced the number of styles detected which made the project feel slightly lacklustre. In order to combat this, the serve style was added as it also occurred frequently throughout a badminton game video.

Last but not least, the initial proposal made the need for a Graphical User Interface (GUI) optional as it was thought that users can simply run the code through the terminal. This thinking came from the team noticing that TrackNetv2 had made users utilise the terminal to get their code running so it was figured that this form of interaction would uphold when it comes to the project's own code. But, as the team tested the program more from the user's perspective, it was obvious that using the terminal alone could be tedious and difficult to understand to some. As such, the Tkinter library was studied and used to build a simple GUI alongside the user guide provided to improve user experience.

Thus, every single deviation from the initial proposal up until the current state of the project had come with purpose to help overcome the minor shortcomings that were discovered for the duration of the project. In fact, having these changes actually improved the overall functionality of the program even though it has not been written in the proposal due to lack of understanding in the field or simply overlooked at such an early stage. So while there were changes, they did not make any major impact on the status of the project, instead, the project was still produced as it was proposed with better performance.

7.0 Conclusion

In conclusion, the completion of this final year project has been a momentous achievement for the team signifying an important milestone in our academic journey. With such a challenging project topic and even when the team had no prior experience in computer vision or machine learning, we showcased our dedication to solving the problem by employing our collective knowledge and skills. Throughout this collaborative endeavour, we have delved into advanced concepts, technologies, and methodologies, acquiring invaluable experience and insights that will shape our future as software developers.

In marking the outcome of this project as a major success, we have achieved our objectives and fulfilled our goals. The opportunities and knowledge gaps that presented themselves while researching this project topic have been bridged with the development of our badminton shot styles analysis tools. This result marks the culmination of months of analysis, design, development, testing, and documentation aided by industry-standard software engineering practices and methodologies such as the Agile framework. Together, the team has demonstrated the ability to collaborate effectively and successfully deliver innovative solutions to real-world problems.

The team hopes to further improve the software by eliminating the limitations that it is currently facing and enriching its features with more statistical insights. The team pledges its commitment to making this software open-source and catering towards the target users who are independent researchers and badminton enthusiasts. Through our collective efforts, we aspire that our software contributes to the advancement and accessibility of sports analytics and fosters a greater understanding of player performances in the context of badminton.

8.0 References

Baodong, Y. (2014). Hawkeye technology using tennis match. *www.academia.edu*. https://www.academia.edu/34589713/Hawkeye_technology_using_tennis_match.

Chen, B. and Wang, Z. (2007). A statistical method for analysis of technical data of a badminton match based on 2-D seriate images. doi: 10.1016/S1007-0214(07)70138-4.

Chu, W. -T. and Situmeang, S. (2017). Badminton Video Analysis based on Spatiotemporal and Stroke Features. doi: 10.1145/3078971.3079032.

Gangal, S., & Raje, S. (2014). The HAWKEYE Technology Shantanu Gangal1 Sangram Raje. *Kuk*. https://www.academia.edu/4325931/The_HAWKEYE_Technology_Shantanu_Gangal1_Sangram_Raje.

Hsu, T., Wang, C., Lin, Y., Chen, C., Jut, N. P., Ik, T., Peng, W., Wang, Y., Tsengt, Y., Huang, J., & Ching, Y. (2019). *CoachAI: A Project for Microscopic Badminton Match Data Collection and Tactical Analysis*. https://doi.org/10.23919/apnoms.2019.8893039

Huang, Y. C., Liao, I. N., Chen, C. H., İk, T. U. and Peng, W. C. (2019). TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications. doi: 10.1109/AVSS.2019.8909871.

Jin Qiu. (2022). Study on the Technical Characteristics of Badminton Players in Different Stages through Video Analysis. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *13*(10), 553–558.

Liao, W. C., Chia-Smith, Y. D., Wu, K. C., Lin, C. C., & Liao, L. A. (2022). Location, Location, Location: Understanding the Impact of Shot Placements in Badminton Men's Double at 2022 World Championship. *Sports Injuries & Medicine*, *6*(5). https://doi.org/10.29011/2576-9596.100192

Lin, K. C., Wei, C. W., Lai, C. L, Cheng, I. L. and Chen, N. S. (2021). Development of a badminton teaching system with wearable technology for improving students' badminton doubles skills. doi: 10.1007/s11423-020-09935-6.

Mahmood, Z., Ali, T., Khattak, S., Hasan, L. and Khan, S. U. (2014). Automatic player detection and identification for sports entertainment applications. doi: 10.1007/s10044-014-0416-4.

Rahmad, N. A., Sufri, N. A., Muzamil, N. H. and As'ari, M. A. (2018). Badminton player detection using faster region convolutional neural network. doi: 10.11591/ijeecs.v14.i3.pp1330-1335.

Su, Y. and Liu, Z. (2018). Position Detection for Badminton Tactical Analysis based on Multi-person Pose Estimation. doi: 10.1109/FSKD.2018.8686917.

Tan, D. Y. W., Ting, H. Y. and Lau, S. B. Y. (2016). A review on badminton motion analysis. doi: 10.1109/ICORAS.2016.7872604.

Wang, W., Chan, T., Yang, H., Wang, C., Fan, Y., & Peng, W. (2021). Exploring the Long Short-Term Dependencies to Infer Shot Influence in Badminton Matches. In *2021 IEEE International Conference on Data Mining (ICDM)*. https://doi.org/10.1109/icdm51629.2021.00178

Xipeng, Z., Peng, Z. and Yecheng, C. (2022). Research on Badminton Teaching Technology Based on Human Pose Estimation Algorithm. doi: 10.1155/2022/4664388.

Yoshikawa, Y., Shishido, H., Suita, M., Kameda, Y. and Kitaharu, I. (2021). Shot detection using skeleton position in badminton videos. doi: 10.1117/12.2590407.

Yunwei, L., & Shiwei, J. (2019). Video Analysis Technology and Its Application in Badminton Sports Training. *Journal of Physics*. https://doi.org/10.1088/1742-6596/1213/2/022009

9.0 Appendix

Sample Source Code

The code below shows our main analyse function which initiates a MediaPipe instance to display player skeleton and runs our prediction model to annotate shot styles on video. It satisfies all shot-style prediction requirements and left-handed player analysis requirements.

The function has been redacted as it exceeds more than 100 lines.

```
def analyse(videoToCapture, output_directory, isLeftHanded = False):
    # read video
    cap = cv2.VideoCapture(videoToCapture)

    # information for output
    fps = int(cap.get(cv2.CAP_PROP_FPS))
    output_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    output_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    new_output = output_directory + videoToCapture.split('/')[-1].split('.')[0] +
"_output.mp4"
    output_video = cv2.VideoWriter(new_output, fourcc, fps, (output_width,
output_height))
    crop_rate = 5

    # Mediapipe Instance
    with mp_pose.Pose(static_image_mode=False, min_detection_confidence=0.5,
min_tracking_confidence=0.5) as pose:
        last_pose = "Unknown"
        while True:
            ret, frame = cap.read()

            if ret:
                # Recoloring Image
                image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                image.flags.writeable = False

                # Make Detection
                results = pose.process(image)

                # Recolor back
                image.flags.writeable = True
                image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

                # Extract

                try:
                    # Check if left handed
                    if isLeftHanded:
                        cv2.flip(image, 1) # flip image horizontally
                    # Get all landmarks
```

```
                    landmarks = results.pose_landmarks.landmark
                    left_shoulder =
[landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,

landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
                    right_shoulder =
[landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].x,

landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value].y]
```

...... REDACTED .....

```
                    # Calculate all Angles
                    left_arm = calculateAngle(left_shoulder, left_elbow,
left_wrist)
                    right_arm = calculateAngle(right_shoulder, right_elbow,
right_wrist)

                    left_leg = calculateAngle(left_hip, left_knee, left_ankle)
                    right_leg = calculateAngle(right_hip, right_knee, right_ankle)

                    left_under_arm = calculateAngle(left_elbow, left_shoulder,
left_hip)
                    right_under_arm = calculateAngle(right_elbow, right_shoulder,
right_hip)


                    if isLeftHanded:
                        cv2.flip(image, 1)

                    # Create dataframe for prediction
                    lst = np.array(
                        [left_arm, right_arm, left_leg, right_leg, left_under_arm,
right_under_arm]).flatten()
                    lst = list(lst)
                    x = pd.DataFrame([lst])
                    model_class = model.predict(x.values)[0]
                    model_prob = model.predict_proba(x.values)[0]

                    if (round(model_prob[np.argmax(model_prob)], 2) > 0.5):
                        # Display Class
                        cv2.putText(image, model_class, [25, 25],
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
                                    cv2.LINE_AA)
                    else:
                        cv2.putText(image, "Unknown", [25, 25],
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
                                    cv2.LINE_AA)

                    # Display Probability
                    cv2.putText(image, str(round(model_prob[np.argmax(model_prob)],
2)), [35, 45],
```

```
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)
                except:
                    pass

                # Render Detection
                mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS)

                # draw preview and write to output
                output_video.write(image)
                cv2.imshow('MediaPipe', image)

                # use Q to terminate
                if cv2.waitKey(10) & 0xFF == ord('q'):
                    break
            else:
                break
        # terminate program
        output_video.release()
        cap.release()
        cv2.destroyAllWindows()
```

The analyse() function is called upon the click of begin button with all input and output paths fed in as arguments

```
# begin button logic
def begin():
    leftHandBool = 0
    filename = input_entry.get()
    output_directory = output_entry.get()
    output_directory += "/"
    if filename != '' and output_directory != '':
        if Checkbutton1.get() == 1:
            leftHandBool = 1
        analyse(filename, output_directory, leftHandBool)  # start mediapipe
```

10.0 Team Members' Contribution Annex

Name	Student ID	Overall Contribution
Matin Raj Sundara Raj	32124260	33.33 %
Louis Juliano	31507794	33.33 %
Suah Wen Hung	30721083	33.33 %

Member	Tasks Done
Suah Wen Hung	1. Software deliverables 2. Software qualities 3. Critical discussion of Software Project as a whole
Louis Juliano	1. Project Outcomes 2. Software limitations & future improvements 3. Software methodology & design
Matin Raj Sundara Raj	1. Introduction & Project Background 2. Updated Literature review 3. Conclusion 4. Sample Source Code 5. Front cover, Appendix, References, style & formatting 6. Compilation & proofreading