USER GUIDE

MCS13

FIT 3162

MEMBERS

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

BadminSmash - Visual Analysis of Badminton Shot Styles

Table of Contents

1.0 End-User Guide

1.1 Software features

This software utilises a machine learning model to identify and classify badminton shot styles of the following categories:

- Serve
- Backhand
- Net Drop
- Forehand (includes forehand smashes)

This software allows the detection and shot classification of a single player standing in front of the net (closer to the camera, viewed directly parallel facing the net). By default, this software analyses and classifies right-handed players. If the player is left-handed, users are to tick the 'left-handed' checkbox in the Input GUI. This software will process the badminton video in the background to annotate the shuttlecock and its trajectory first. Then, a preview window will appear that shows live shot style detection and annotation. Then, the output video is stored in the directory specified by the user.

1.2 How to use the program

The step-by-step process to operate the software is as follows:

To Run Program

1. Open Command Prompt (Windows Users) or Terminal (MacOS)
2. Copy the full pathname of the 'prediction' folder and change the directory into the 'prediction' folder using the <cd> command
3. Run the compile.py file using <python3 compiled.py> command
4. A screenshot of the terminal having done the steps above is shown in Diagram 1 below

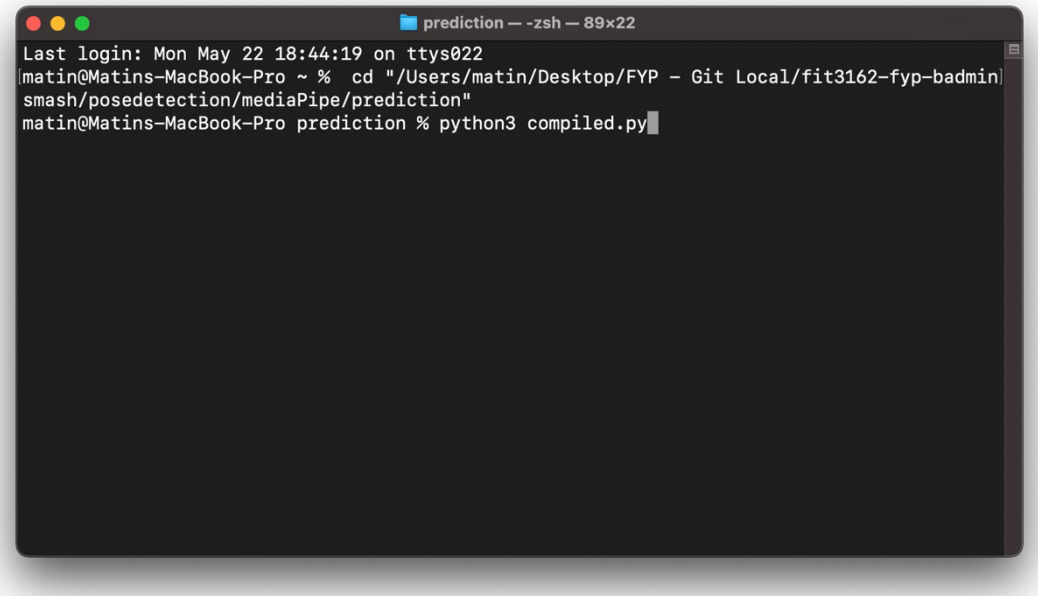

Diagram 1: Commands to run program

Input GUI

Diagram 2: Input GUI

1. Upon running the compiled.py file, an Input Graphic User Interface (GUI) will appear. Refer to Diagram 2 above
2. Click the checkbox if the player to be analysed in the video is left-handed. Otherwise, the software will assume that the player is right-handed by default
3. Click the Browse button in the 'Input File Path' section. A file explorer window will appear. Choose a .mp4 video to be analysed
4. Click the Browse button in the 'Output File Path' section. A file explorer window will appear. Choose an output directory to store the output video
5. Click on 'Begin Analysis' button

Background Processing

1. The video will first track the shuttlecock in the background. You will see a "Tracking shuttlecock…" message on the terminal. Once it is done, a "Shuttlecock tracking done" message is printed on the terminal.
2. After that, the shuttlecock trajectory will be processed in the background. You will see a "Processing shuttlecock trajectory..." message on the terminal. Once it is done, "Shuttlecock trajectory processed" will be printed on the terminal.
3. The terminal will be in the following state as shown in Diagram 3

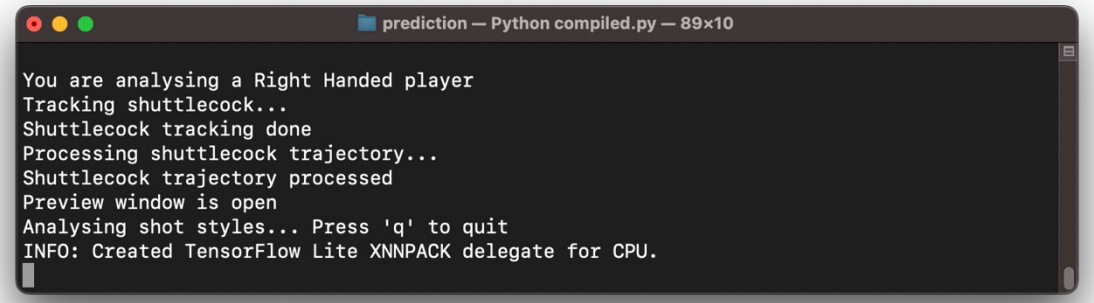

Diagram 3: Background processing messages

Output

1. A preview window will be opened to show live analysis and annotation of badminton shots classifications for the player
2. You may wait until the whole video is played finish or press ‘q’ on the keyboard to interrupt the analysis.
3. An output video with the shot styles annotation (up till the interrupt point) is saved in the directory specified earlier.

To Exit Program

1. Click on the ‘Quit’ button on the Input GUI to exit the program

1.3 Interpreting results

Diagram 4: Background processing messages

Diagram 4 above shows a screenshot of the preview window. A skeleton is printed onto the player and follows their movement. The current shot style of the player is annotated in the top left corner of the video. For example, in Diagram 4, the player is currently performing a serve. The value below the shot style indicated the confidence rate ranging from [0..1]. For example, in Diagram 4, the software is 83% confident that the player is performing a serve. The shuttlecock is tracked with a red dot. The shuttlecock's trajectory is tracked with markers based on location.

1.4 Error messages

Error messages will be printed onto the terminal

Error code 0: The program has crashed unexpectedly. Please re-run the program. Ensure you have all the necessary libraries installed properly.

Error code 1: Error reading input video. Please make sure the input video path exists and the video is in .mp4 format

Error code 2: Error saving output video to the specified directory. Please ensure the output directory given is valid

2.0 Technical Guide

2.1 Installation

Installing Keras and TensorFlow through Anaconda

To install Anaconda, follow this link to bring you to the official Anaconda download page. Follow the steps on downloading Anaconda until you are able to start the Anaconda Navigator.

1. Start up your Anaconda Navigator (as shown in Diagram 5 below)

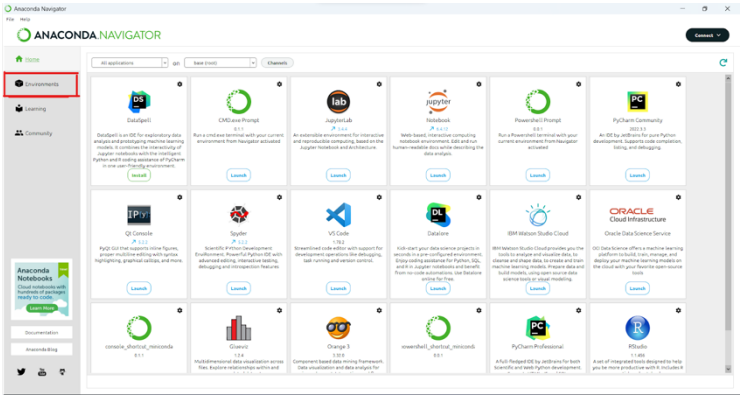

Diagram 5

2. Click on Environment (as shown in Diagram 6 below)

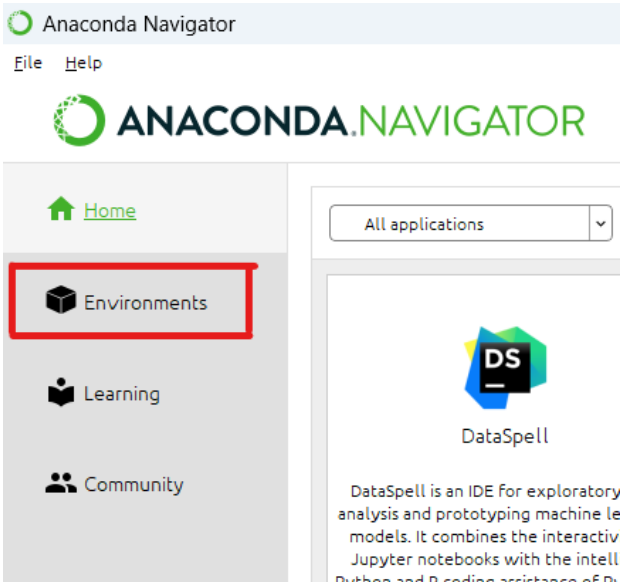

Diagram 6

3. Click on Create (as shown in Diagram 7 below)

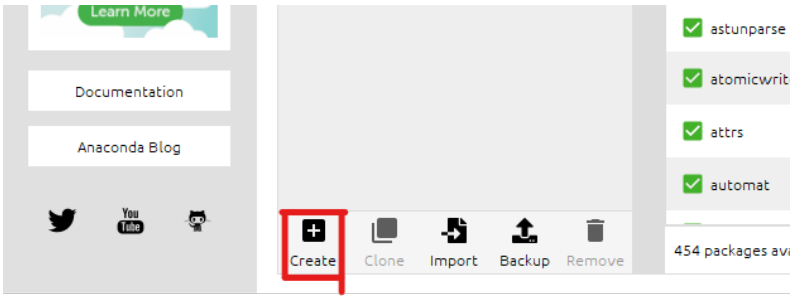

Diagram 7

4. Name your environment and tick on Python and set the python version you are using (as shown in Diagram 8 below)

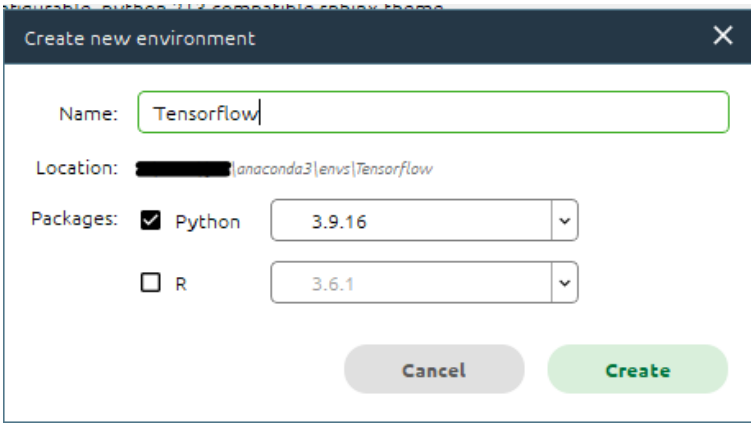

Diagram 8

5. Then click on create
6. At the right top, click on installed package and search for Tensorflow
7. Tick both Tensorflow and Keras (as shown in Diagram 9 below)

Diagram 9

8. Then click on Apply and wait for the installation
9. After installing, click on apply and wait for it to properly install all the dependencies
10. Your TensorFlow environment is ready to go.

NB: If you are using Anaconda in VS Code, make sure that it is running the correct Python interpreter and the correct Conda environment.

Installing Other Necessary Libraries

To install the other libraries needed to run the program:

1. Open the Conda environment that you have set up. Start a new terminal
2. Copy the full directory path of where the requirements.txt file is located in the project folder.
3. Move into the directory containing the requrements.txt file using the <cd *[path]*> command on the terminal
4. Enter the command <pip3 install -r requirements.txt> into the terminal
5. Your libraries have been installed

2.2 Source Code Manipulation

The provided source code can be manipulated manually to achieve the desired result. This requires a little more understanding of where to locate and identify certain functions and variables. For example, to allow video analysis to run faster or to disable shuttlecock tracking & trajectory, the TrackNetV2 portion of the code can be disabled. Besides that, to adjust the confidence rate of pose detection, the min_detection_confidence variable can be changed.

Further information regarding this can be found in readMe.txt