Provoking High-Ball Returns in Table Tennis: Analyzing Gameplay with TTNet

Ayush Rawal, Niket Mittal, Prakhar Kesari

Abstract

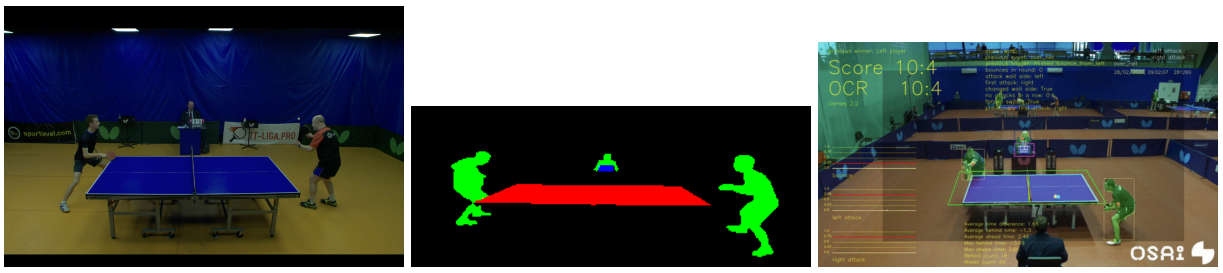
In competitive table tennis, provoking high-ball returns offers a strategic advantage, enabling offensive plays such as smashes. This study explores how specific shot types and placements influence opponents to produce high-ball returns. Leveraging TTNet, a multi-task deep learning model for real-time video analysis, the research introduces enhancements to classify ball height and identify weak returns. Using the OpenTTGames dataset—featuring high-frame-rate annotated match footage—we perform ball detection, event spotting, and semantic segmentation to analyze spatial and temporal patterns in gameplay. Our methodology computes ball speed, estimates z-coordinates, and visualizes shot distributions, ultimately extracting insights that could inform player strategies. The results pave the way for predictive analytics in table tennis, enabling tactical decision-making based on real-time video data.

Keywords: Classification, Computer Vision, CNN

1 Introduction

In competitive table tennis, the ability to identify and exploit weak returns is critical for gaining a strategic advantage. One particularly advantageous return is the high ball, which allows the player to execute a powerful smash. Understanding how to provoke high-ball returns from an opponent requires in-depth analysis of shot patterns, ball trajectories, and player responses. With the advent of advanced computer vision models such as TTNet[?], there is an opportunity to systematically analyze and identify the precise conditions that lead to high returns.

TTNet is a multi-task deep learning model for real-time temporal and spatial video analysis of table tennis. It performs ball detection, event spotting (such as bounce and net hits), and semantic segmentation to identify key objects like the player, table, and scoreboard. This research proposes extending TTNet to classify ball height and extract actionable insights on how specific shot types and placements can provoke high-ball returns. By modifying the model to capture these patterns, we aim to offer strategic guidance for players to create more offensive opportunities.



(a) Snapshot of a Match (b) Annotation of a Match (c) Analysis of Game

Figure 1: Extracting features for the dataset from a match of Table Tennis.

2 Motivation

The primary motivation of this work is to empower table tennis players with data-driven strategies to enhance their performance. While elite players often rely on intuition and experience to exploit opponents' weaknesses,

systematic analysis through machine learning can reveal patterns and strategies that are not immediately apparent. Specifically, by understanding how to force high-ball returns, players can increase their chances of executing smashes, which are among the most effective point-winning shots in table tennis.

This research also bridges the gap between sports analytics and real-time video analysis, providing a framework for understanding opponent behavior. Beyond competitive play, the insights derived from this model can also be used for training, allowing players to simulate and practice scenarios that lead to high returns. Additionally, such analysis can benefit coaches by providing empirical evidence to tailor personalized strategies for their athletes.

3 Use Cases and Stakeholders

The proposed system can have a significant impact on various stakeholders:

1. Players: Gain insights into the types of shots and strategies that increase the likelihood of high returns, enabling them to adopt a more aggressive playing style.

2. Coaches: Use empirical data to design training regimens that simulate scenarios leading to high-ball opportunities.

3. Analysts and Researchers: Study game dynamics in greater detail, facilitating the development of new tactics and strategies.

4. Broadcasters and Spectators: Enhance the viewing experience by providing real-time insights into player strategies and shot effectiveness.

Applications include personalized coaching programs, automated feedback systems for training, and live analytics for professional matches. However, the system has limitations such as the need for high-quality video input, potential inaccuracies in rapid ball movement, and the computational demands of real-time processing.

By building upon the existing TTNet framework, this research not only advances the technical capabilities of sports analytics but also provides practical benefits for improving competitive performance in table tennis.

4 Related Work

Previous research on table tennis analytics has primarily focused on ball tracking and event detection. TTNet itself is a landmark contribution due to its ability to perform multi-task learning and deliver real-time inference at over 120 frames per second.

Existing works on tactical analysis in sports primarily rely on manual annotation or basic statistical approaches. For instance, studies on tennis have explored how specific shot placements influence opponent responses, but these analyses typically lack real-time automation. In the table tennis domain, recent research has leveraged pose estimation[?] and trajectory tracking with spin[?] for skill assessment, but there remains a gap in predictive modeling for opponent behavior.

By extending TTNet to classify ball height and correlate it with preceding shots, we address this gap and provide actionable insights. Similar multi-task models in other sports, such as soccer and basketball, have shown that combining spatial and temporal data can improve tactical decision-making. Our approach builds on these findings by integrating ball height classification and shot pattern analysis to identify conditions that provoke high-ball returns.

Data Sources and TTNet Exploration for Ball Speed and Weak Shot Classification

---

Ayush Rawal (2024201036), Niket Mittal (2024201076), Prakhar Kesari (2024202023)

1. Introduction

The analysis of table tennis ball movement has become an integral part of sports analytics, with deep learning techniques playing a crucial role in tracking, event detection, and predictive modeling. In this study, we aim to compute the speed of the ball, estimate its z-coordinate at the edges of the table, and classify weak shots based on predefined height thresholds. To achieve these objectives, we leverage TTNet, a multi-task deep learning framework, and the OpenTTGames dataset, which provides high-speed annotated table tennis video recordings. This paper details the data sources, extraction processes, and initial exploration of the dataset.

2. Data Sources

2.1 OpenTTGames Dataset

The OpenTTGames dataset serves as the primary source of data for this study. The dataset consists of:

- **High-speed video recordings**: Full HD (1920×1080) resolution at 120 frames per second (fps).
- **Ball position annotations**: Frame-wise (x, y) coordinates of the ball.
- **Event labels**: Key game events such as ball bounces and net touches.
- **Semantic segmentation masks**: Labels for players, table, and scoreboard.

Relevance to This Study

- The high frame rate enables precise ball tracking and velocity computation.
- Event annotations allow the identification of bounce points, critical for z-coordinate estimation.
- Semantic segmentation aids in filtering out occlusions caused by players or background elements.

3. Data Processing

3.1 Image Extraction and Preprocessing

To facilitate analysis, individual frames are extracted from the videos using OpenCV and FFmpeg while preserving the original frame rate (120 fps). Each frame undergoes preprocessing, including:

- **Resizing** (if necessary) to match TTNet's input requirements.
- **Normalization** to enhance contrast and eliminate lighting inconsistencies.

3.2 Ball Detection

TTNet employs a two-stage detection mechanism to locate the ball:

1. **Global detection**: A low-resolution version of the frame is used to coarsely localize the ball.
2. **Local detection**: A high-resolution crop around the global detection result refines the ball's coordinates.

Final ball coordinates (x, y) are computed using:

$$x = x_1 \frac{w_0}{w_1} - \frac{w_2}{2} + x_2$$

$$y = y_1 \frac{h_0}{h_1} - \frac{h_2}{2} + y_2$$

Where x_1, y_1 and x_2, y_2 are the global and local detections, and w, h represent width and height parameters at various processing stages.

3.3 Event Spotting

The event spotting module in TTNet identifies game events based on temporal frame sequences:

- **Probability-based event detection** assigns likelihoods to occurrences such as bounces and net hits.
- **Sliding window techniques** improve accuracy by considering adjacent frames for context.

Detected events provide reference points for trajectory analysis and z-coordinate estimation.

3.4 Semantic Segmentation

Semantic segmentation plays a crucial role in enhancing analysis accuracy:

- **Table segmentation** ensures precise identification of valid bounce zones.
- **Player and referee masking** reduces false ball detections due to occlusions.
- **Scoreboard recognition** assists in cross-referencing match progress with game analytics.

4. Data Exploration

4.1 Ball Position Distribution

To assess the dataset's usability, we analyze the spatial distribution of ball positions:

- **Heatmaps** illustrate areas where the ball appears most frequently.
- **Trajectory clustering** helps identify typical shot patterns.

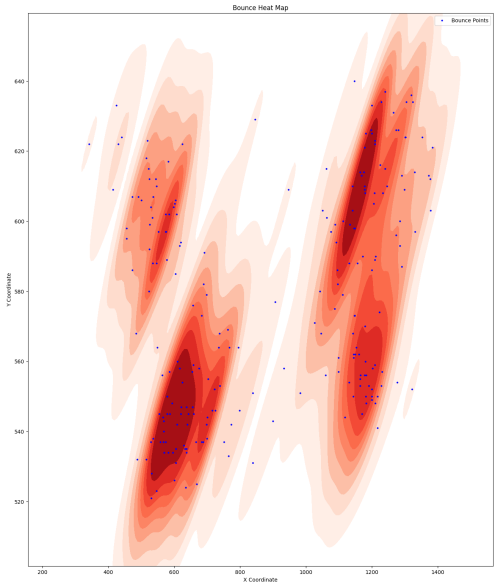


Figure 1: Heat map of ball bounce locations

4.2 Speed Analysis

Velocity is computed based on frame-wise displacement using the formula:

$$v = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{\Delta t}$$

where $\Delta t = 1/120$ seconds (frame interval). Key findings include:

- Ball speeds typically range from 4 to 15 m/s.

- Bounces result in speed reductions of around 8-10%.
- High-speed shots correlate with smashes and aggressive strokes.

Figure 2: Speed Analysis

5. Conclusion

This study leverages the OpenTTGames dataset and TTNet framework to analyze ball dynamics in table tennis. Through multi-task processing—including ball detection, event spotting, and semantic segmentation, we identify valuable features for speed computation and weak shot classification. Future work will focus on finding z-coordinate estimations and identifying other relevant features for improved estimations.

Tennis Shot Height Classification

Abstract

This report presents a machine learning system designed to classify tennis shots into height categories (low, medium, high) based on ball trajectory data extracted from match videos. By utilizing a neural network architecture with carefully selected features including court coordinates, pre-bounce velocity, and bounce position coordinates, the system achieves 78% overall accuracy. The research demonstrates the feasibility of automated shot height classification in tennis analytics, providing a framework for advanced match analysis, player performance evaluation, and tactical assessment. The model's robust performance across different height categories suggests potential applications in real-time match analysis, coaching assistance tools, and broadcast enhancements.

1. Introduction

Tennis analytics has evolved significantly in recent years, with increasing emphasis on quantitative measures to understand player performance, strategy, and match dynamics. Among various shot attributes, height plays a crucial role in determining tactical advantage, yet remains challenging to quantify systematically during matches. This project addresses this challenge by developing an automated system for classifying tennis shots into three distinct height categories:

- **Low**: Shots that maintain a trajectory close to net height
- **Medium**: Shots that achieve moderate elevation above the net
- **High**: Shots with significant vertical trajectory above the net

Shot height classification provides valuable insights into player strategy, court positioning, and tactical patterns. High shots typically create defensive positions, while low shots can be aggressive or defensive depending on court position and spin.

This research leverages machine learning techniques to analyze ball trajectory data from tennis match videos, focusing on the relationship between pre-bounce velocity vectors, bounce coordinates, and resulting shot height categories.

2. Related Work

While tennis analytics has seen advancements in areas such as player tracking, stroke identification, and tactical analysis, shot height classification remains relatively underexplored in the literature. Existing approaches to tennis video analysis have focused primarily on:

- Court detection and camera calibration techniques
- Ball tracking algorithms in video footage
- Player movement and positioning analysis
- General shot type classification (forehand, backhand, serve)

Some related works are:

1. Voeikov, R., Falaleev, N., & Baikulov, R. (2020). TTNet: Real-time temporal and spatial video analysis of table tennis. *arXiv preprint arXiv:2004.09927*. https://arxiv.org/abs/2004.09927
2. Wu, T. (n.d.). Camera-Based Table Tennis Posture Analysis. https://wutonytt.github.io/Camera-Based-Table-Tennis-Posture-Analysis/
3. Achterhold, J., Tobuschat, P., Ma, H., Buechler, D., Muehlebach, M., & Stueckler, J. (2023). Black-Box vs. Gray-Box: A Case Study on Learning Table Tennis Ball Trajectory Prediction with Spin and Impacts. *arXiv preprint arXiv:2305.15189*. https://arxiv.org/abs/2305.15189

This project builds upon these foundational elements, particularly court mapping through perspective transformation and ball tracking, while introducing specialized classification for the height dimension of shots.

3. Methodology

3.1 Data Collection and Processing

The dataset for this study was compiled from five professional tennis matches, processed through the following pipeline:

1. **Court Calibration**: Court corners were marked to establish a coordinate system through perspective transformation
2. **Ball Tracking**: Ball positions were extracted for each video frame using the OpenTTGames dataset
3. **Event Detection**: Bounce events were identified and annotated
4. **Feature Engineering**: Velocity vectors and transformed court positions were calculated
5. **Height Categorization**: Shots were labeled based on trajectory analysis

The OpenTTGames dataset was instrumental to this research, providing high-quality ball tracking data from professional tennis matches. This dataset includes frame-by-frame ball position coordinates and events, enabling accurate velocity calculations.

3.2 Dataset Generation Process

A critical component of this research was the development of a robust dataset generation pipeline, implemented in the `generate_dataset.py` script. This process involved:

3.2.1 Perspective Transformation Court coordinates were standardized by applying a homography-based perspective transformation using OpenCV's `findHomography` function. This crucial step mapped pixel coordinates from video frames to a standardized court coordinate system in meters, enabling consistent spatial analysis across different camera angles and match footage.

3.2.2 Bounce Detection and Analysis For each detected bounce event, the system:

1. Located the exact bounce position in transformed court coordinates
2. Calculated pre-bounce velocity vectors using a 60-frame window before bounce
3. Applied sophisticated post-bounce trajectory tracking to determine height outcomes
4. Used geometrically defined break lines to categorize shot heights consistently

3.2.3 Height Classification Algorithm Shot height categorization employs a sophisticated physics-based approach using three parallel break lines positioned at carefully calibrated heights above the court. The algorithm determines height categories by analyzing the ball's position relative to these break lines:

1. **Break Line Definition**: Three parallel lines are established at different heights above the court surface, with distances calculated based on the net post height. These lines create distinct regions in the vertical space above the court.
2. **Dynamic Break Line Selection**: The system intelligently selects appropriate break lines based on which side of the court the bounce occurs, using the court's midpoint as the decision boundary. This adaptation ensures consistent height categorization regardless of bounce location.
3. **Post-Bounce Trajectory Analysis**: Rather than simply using the bounce position, the algorithm tracks the ball's complete post-bounce trajectory to identify the maximum height reached, which provides a more accurate representation of the shot's true height.
4. **Velocity Direction Detection**: The system monitors vertical velocity changes to precisely identify when the ball reaches its peak height after bounce. This enables accurate height classification even for complex trajectories.
5. **Interval-Based Classification**: The algorithm uses a mathematical approach to determine which interval (between break lines) contains the ball's trajectory peak, mapping these intervals to the three height categories: low, medium, and high.

This refined approach ensures robust height classification across various court positions and camera angles, addressing challenges related to perspective distortion and variable bounce characteristics.

3.2.4 Dataset Compilation Individual match datasets were processed independently and then merged using a specialized tool (`merge_csv.py`) that ensured consistent field ordering and data validation. This approach enabled incremental dataset expansion while maintaining strict quality control.

It's important to note that despite the sophisticated algorithms implemented, the dataset generation process is not fault-free. The height classification algorithm can occasionally generate incorrect height categories due to several factors:

1. Imperfect ball tracking data in the source footage
2. Challenges in accurately detecting velocity direction changes in complex trajectories
3. Edge cases where the ball's path crosses multiple break lines in rapid succession
4. Perspective distortion effects not fully compensated by transformation

These limitations are reflected in the final classification accuracy and contribute to the challenges in distinguishing between medium and high shots in particular.

The final compiled dataset exhibits a distribution of approximately 49% low shots, 19% medium shots, and 32% high shots, providing sufficient representation across all categories despite the noted limitations.

3.3 Feature Selection

After experimental analysis, twelve key features were selected as inputs for the classification model:

- Court corner coordinates (tx1, ty1, tx2, ty2, tx3, ty3, tx4, ty4) for spatial context
- Pre-bounce velocity in x-direction (vx_before)
- Pre-bounce velocity in y-direction (vy_before)
- Bounce x-coordinate in court reference frame (bounce_x)
- Bounce y-coordinate in court reference frame (bounce_y)

This feature set provides comprehensive spatial context along with the bounce position, enabling the model to better interpret the trajectory relative to the court geometry. By incorporating court corner coordinates, the model can learn spatial relationships specific to different areas of the court, improving its ability to classify shot heights accurately across various bounce locations.

3.4 Model Architecture

The classification model (HeightNet) implements a feedforward neural network with the following structure:

- Input layer: 12 neurons (corresponding to the selected features)
- First hidden layer: 64 neurons with ReLU activation, batch normalization, and dropout (0.25)
- Second hidden layer: 32 neurons with ReLU activation
- Output layer: 3 neurons (representing low, medium, high categories)

The model was trained using cross-entropy loss and the Adam optimizer with a learning rate of 3×10ˆ-3. Early stopping was implemented with a patience of 3 epochs to prevent overfitting.

4. Implementation

4.1 Data Preprocessing

The implementation includes:

- Feature standardization using StandardScaler
- Train-test split (80%-20%) with stratification to maintain class balance
- PyTorch dataset and dataloader implementation for efficient batch processing

4.2 Training Process

The model was trained for 8 epochs with a batch size of 32, using the following procedure:

1. Forward propagation of batch data
2. Loss calculation using cross-entropy
3. Backpropagation and parameter updates using Adam optimizer
4. Validation after each epoch
5. Model checkpoint saving based on validation accuracy
6. Early stopping if no improvement was observed for 3 consecutive epochs

4.3 Inference Pipeline

For inference on new match data, the system:

1. Processes video frames to extract ball positions
2. Detects bounce events
3. Calculates pre-bounce velocity and bounce position
4. Standardizes features using the saved scaler
5. Forwards the preprocessed data through the trained model
6. Returns the predicted height category with probability scores

5. Results and Evaluation

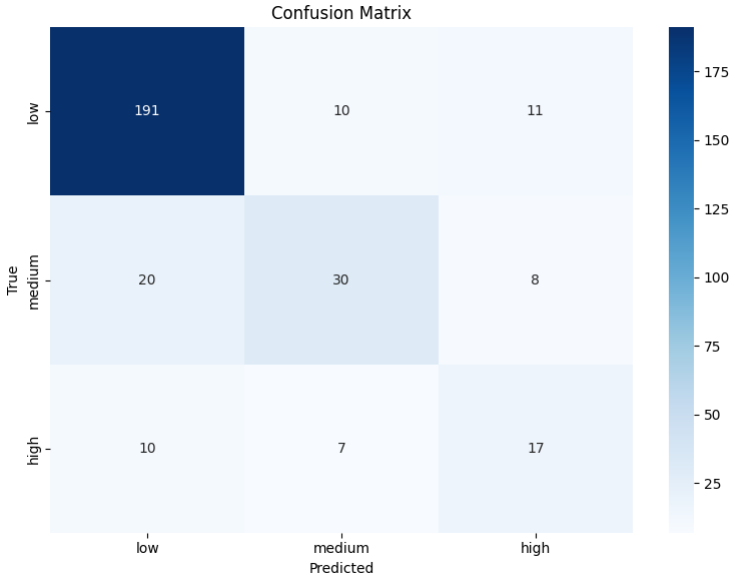
5.1 Model Performance

The final model achieved an overall accuracy of 78% on the validation set, with the following per-class metrics:

Classification Report:

Shot Type	F1-score	Precision	Recall
Low shots	0.88	0.86	0.90
Medium shots	0.57	0.64	0.52
High shots	0.49	0.47	0.50

5.2 Confusion Matrix Analysis



The confusion matrix reveals specific classification patterns:

- Low shots are most accurately classified (191 correct, 21 misclassifications)
- Medium shots show moderate confusion with both low and high categories (30 correct, 28 misclassifications)
- High shots show confusion with both low and medium categories (17 correct, 17 misclassifications)

This suggests that the boundary between categories is not perfectly distinct, with particular challenges in separating medium and high shots. This aligns with the physical intuition of shot trajectories where the transition between height categories can be gradual.

5.3 Training Dynamics

The model showed consistent improvement during training, with validation accuracy increasing from 69.08% in epoch 1 to a peak of 80.59% in epoch 7, with some fluctuations in later epochs. The final model achieved a validation

accuracy of 78.29% in epoch 10. This pattern suggests that while the model learned useful patterns, there may be inherent challenges in the dataset that prevented achieving higher accuracy levels. The fluctuations in validation accuracy (particularly the dip in epoch 6) indicate that the model was sensitive to the specific examples in each validation batch.

6. Discussion

6.1 Feature Importance

The success of the model using only four input features (pre-bounce velocities and bounce coordinates) demonstrates that these parameters capture the essential physics governing shot height. This aligns with the principles of projectile motion, where initial velocity vectors and takeoff point determine trajectory.

6.2 Classification Challenges

The confusion between medium and high shots suggests a continuous rather than discrete transition between these categories in real-world data. This indicates potential for future work in either:

1. Refining the category boundaries with more precise definitions, or
2. Exploring regression-based approaches to predict height as a continuous value

6.3 Practical Applications

This classification system enables several practical applications:

1. **Tactical Analysis**: Identifying patterns in shot height selection based on court position and match situation
2. **Player Development**: Tracking a player's tendency to hit shots of different heights in various scenarios
3. **Opposition Analysis**: Recognizing opponents' height patterns to develop counter-strategies
4. **Broadcast Enhancement**: Providing real-time shot statistics to enhance viewer experience

7. Conclusion

This project successfully demonstrates the viability of automated tennis shot height classification using a neural network approach. The achieved 78% accuracy across three height categories confirms that the selected features and model architecture effectively capture the relationship between ball trajectory parameters and resulting shot height, though there remains room for improvement.

The model architecture utilizing twelve input features, including court corner coordinates, provides a comprehensive spatial understanding that improves classification accuracy. The refined height classification algorithm with dynamic break line selection further improves the system's robustness across different court regions.

The system provides a foundation for advanced tennis analytics by quantifying an important but previously under-analyzed dimension of shot selection. Future work could extend this approach by:

1. Integrating real-time ball tracking for live match analysis which can be done using TTNet
2. Expanding the classification to include additional shot attributes (spin, power)
3. Developing a mobile application for on-court training feedback
4. Combining height classification with tactical positioning data for comprehensive strategy analysis
5. Improving classification accuracy for medium and high shots through additional training data or feature engineering

This research contributes to the growing field of sports analytics by providing a specialized tool for tennis match analysis, with potential applications in coaching, player development, broadcast enhancement, and tactical planning.

Footnote

- GitHub Repo Link

References

1. OpenTTGames dataset - Ball tracking and event detection in tennis matches
2. PyTorch documentation - Deep learning framework
3. OpenCV - Computer vision library for video processing and perspective transformation