May 21, 2025

SMARTRALLY

AI Approach to Coaching Badminton Strategy with Mobile App

Computer Systems Lab

2024-2025

İpek Sayar, Jacob Dipasupil

Problem

Popularity

- No Olympic medals
- “Backyard sport”

Availability

- Majority of badminton clubs are located in California [1]
- Only one club in VA with dedicated badminton facilities [1]
- Not accessible depending on location

Cost

- Badminton coaching costs around $100 per hour
- Not accessible to lower income families

SmartRally

We have developed an AI-powered app for badminton that leverages advanced computer vision and machine learning techniques to enhance players' skills by analyzing their strategy to offer personalized feedback.

Novelty

- Pose recognition
- Next Best Move Algorithm
- Mobile App

We leverage YOLOv11 for object detection, OpenPose for player detection and joint recognition, TrackNet for shuttlecock tracking, and SlowFast for pose classification.

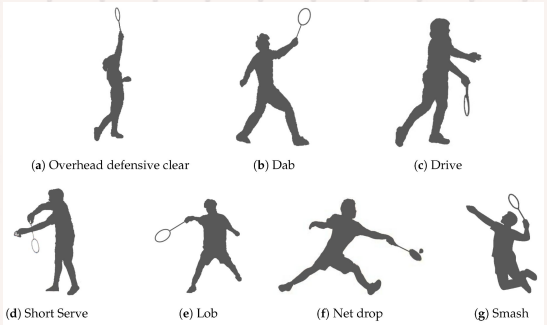
Datasets

VideoBadminton [2]

- Clean, high quality footage of badminton form and technique.
- 18 categories of badminton actions
  - 7,822 clips spanning 145 minutes of self-recorded footage

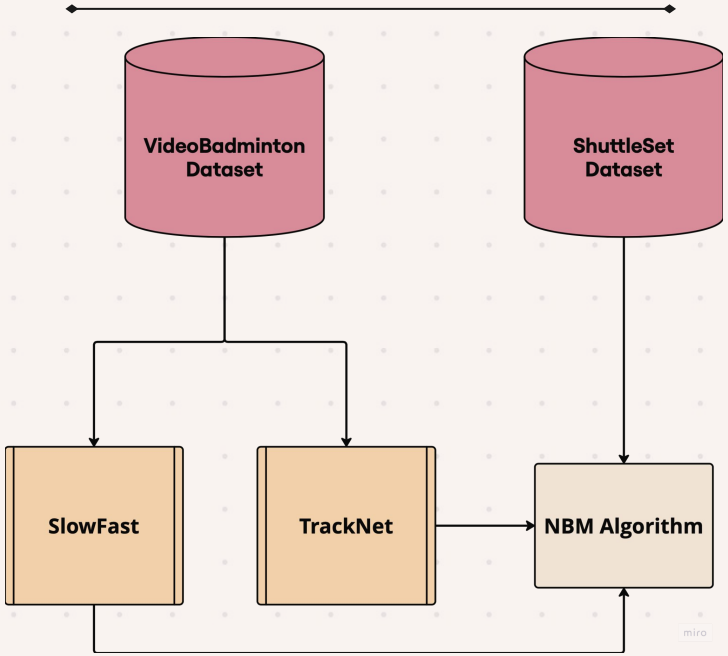
ShuttleSet [4]

- Annotated stroke-level records from matches from 27 top-ranking men's and women's singles players
- Preprocessed: Only kept five key features — shot, hit_area (0–15), land_area (0–15), player_location_area (0–8), and opponent_location_area (0–8) — to use as input for the NBM algorithm.

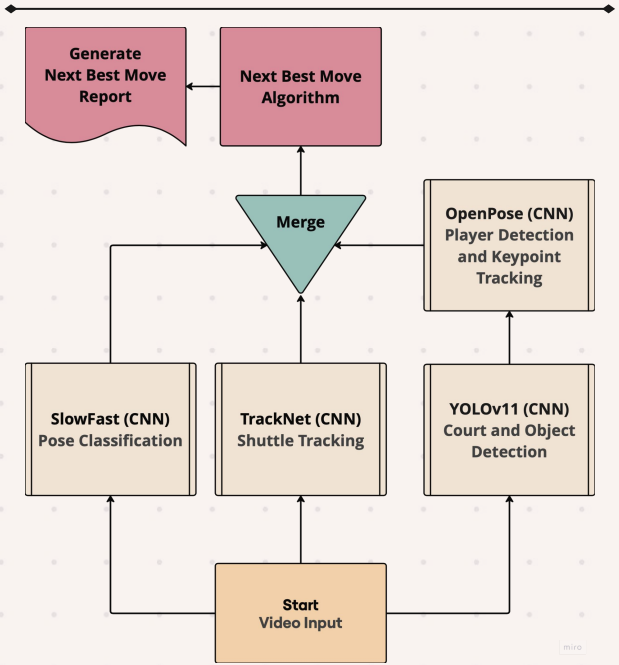


Systems Architecture

Training

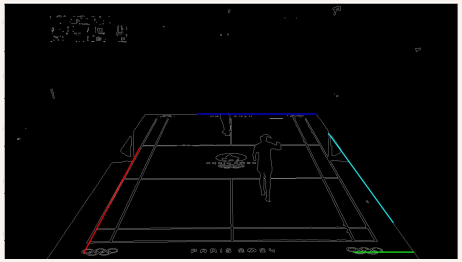


Model Structure



Player Tracking

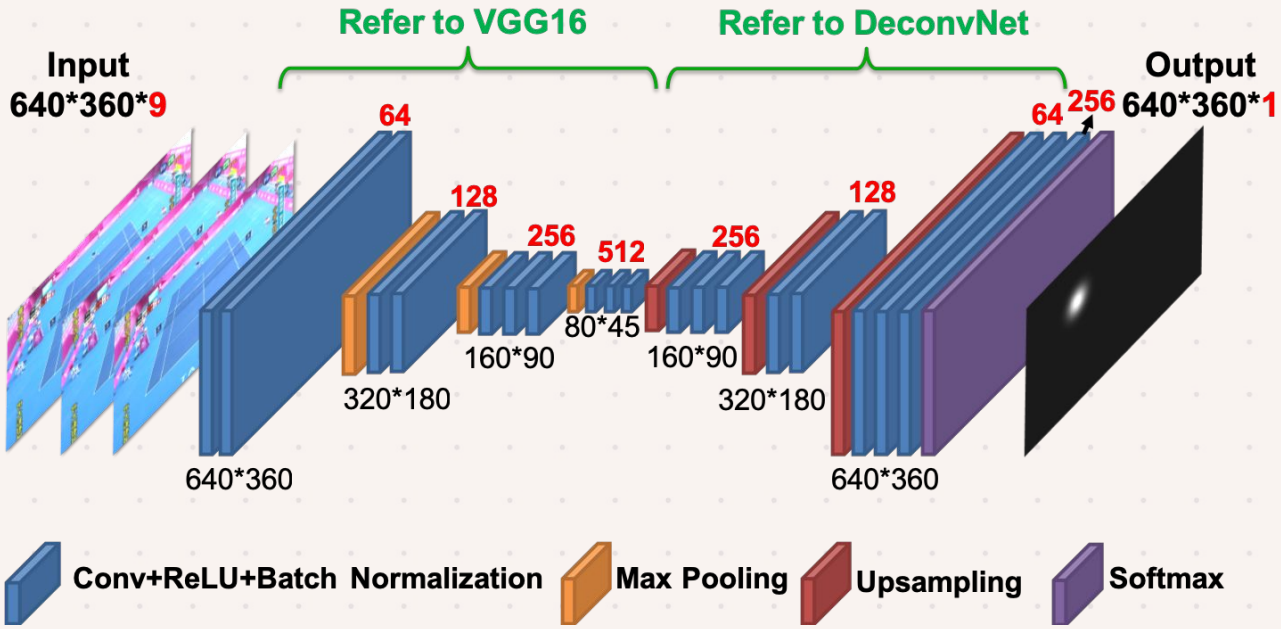
- Each video is split into frames (10-15 frames per second).
- **Canny edge detection** highlights the sharp edges of the court.
- **Hough Line Transform** detects straight lines from the edges to define the court's boundaries and extract coordinates.
- **YOLOv11's OpenPose** processes each frame to detect people and their keypoints.
- Using the court corner coordinates and a y-buffer, we save the keypoints of the players on the court.
- **OpenCV** is used to highlight key points with red circles.
- By focusing on the last two keypoints (the feet), we assign each player a number (0-8) based on which section of the court they are in.



Hough Line Transform

OpenPose Demo

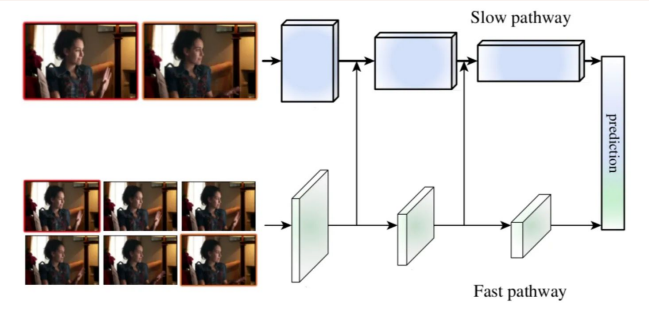
TrackNet [3]

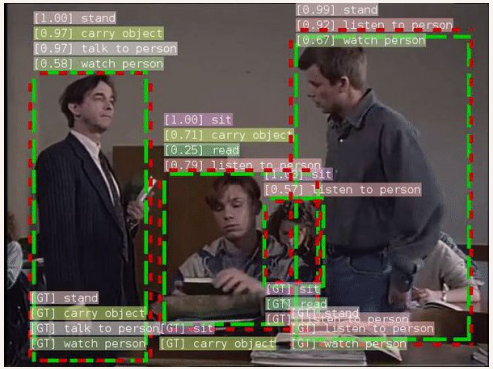

Refer to VGG16
Refer to DeconvNet
Input
640*360*9
Output
640*360*1
64
128
256
512
256
128
64
256
640*360
320*180
160*90
80*45
160*90
320*180
640*360
Conv+ReLU+Batch Normalization
Max Pooling
Upsampling
Softmax

TrackNet Demo

SlowFast[5]

- Slow pathway: Slow, high-definition CNN to analyze the static content of a video
- Fast pathway: fast, low-definition CNN that analyzes the dynamic content of a video.
- Trained on VideoBadminton dataset
  - Achieved Top-1 accuracy of ~72%





Next Best Move (NBM)

1-2 Hidden Layers → Underfitting

- That means the model ends up too simple, it misses the deeper patterns.

>3 Hidden Layers → Overfitting

- Memorizes the training set and isn't able to generalize testing set

ReLU helps vanishing gradients

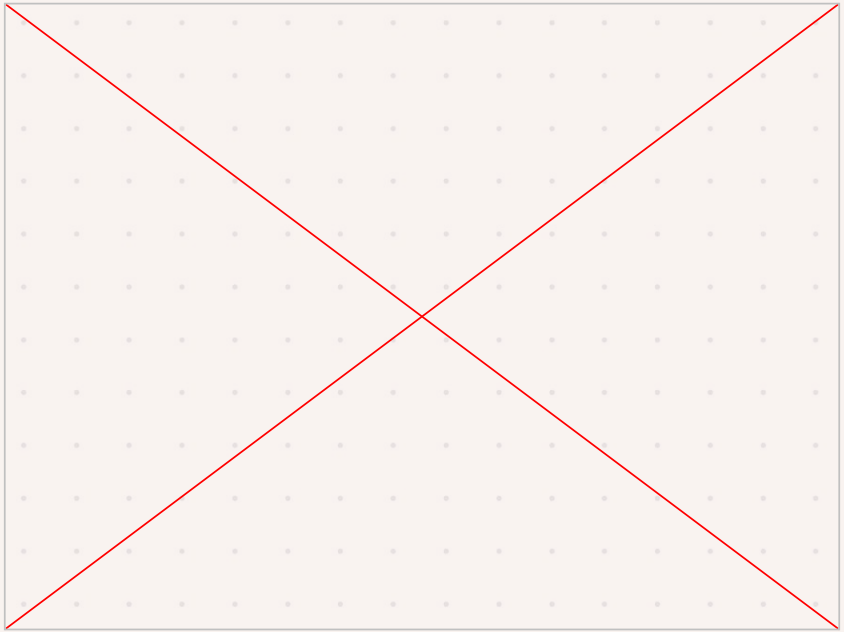
- When learning freezes while the error is still high.

```
nn.Sequential(
# Input: [shot, hit_area (0-15), land_area (0-15), player_location_area(0-8), opponent_location_ar
    nn.Linear(5, 32),
    nn.ReLU(),           # Prevent vanishing gradient problem
    nn.Linear(32, 64),   # Hidden layer 1
    nn.ReLU(),
    nn.Linear(64, 64),   # Hidden layer 2
    nn.ReLU(),
    nn.Linear(64, 32),   # Hidden layer 3
    nn.ReLU(),
    nn.Linear(32, 2)     # Output Layer: [next best move, next best location]
)
```

No one absolute best move in every sequence

- Model gives statistical best choice
- Like a chess engine: guidance, not commands

Mobile App Demo



Conclusion: Developing Game Sense

- Recreate scenarios the app gives you and practice the suggested “next best move”
- Re-record and feed videos back into the app to track how your decision-making evolves
- Build muscle memory and strategy over time
- It's like driving:
  - New drivers know rules, but not instincts
  - Real skills come from reacting to live situations
  - Patterns emerge with repetition
  - Eventually, you just *know* the right move

Questions?

References

[1] https://www.worldbadminton.com/whereToPlay/unitedStates/index.html#va

[2] https://arxiv.org/html/2403.12385v1#S3

[3] https://inoliao.github.io/CoachAI/

[4] https://github.com/wywyWang/ShuttleNet

[5] https://github.com/facebookresearch/SlowFast