Basketball Out-of-Bounds Detection Using Deep Learning

Automated Possession Determination from Game Footage

Abstract

Developed a deep learning system to automatically determine ball possession after out-of-bounds plays in basketball games, achieving 85% accuracy using only 20 training clips. The system combines MobileNetV2 for spatial feature extraction with LSTM networks for temporal modeling, enhanced with jersey colour metadata. This work demonstrates the feasibility of automated officiating assistance to reduce game delays caused by video review.

---

1. Introduction

Problem Statement

Out-of-bounds calls in basketball require referees to identify which player last touched the ball before it crossed the boundary line. When unclear, officials must review video footage at courtside monitors, causing game delays that disrupt momentum and allow players to cool down. Current manual review can take 1-5 minutes per call.

Objective

Build an automated system that can instantly determine ball possession from video clips to eliminate review delays while maintaining high accuracy.

Technical Challenges

- **Fast motion**: Ball and player movements occur in milliseconds
- **Visual occlusion**: Multiple players often obscure the critical moment
- **Jersey similarity**: Team uniforms can look similar in lighting/motion blur
- **Limited data**: Only 20 labeled video clips available for training
- **Different angles**: Each video is shot at a different angle with a different camera

---

2. Methodology

2.1 Dataset

- **Size**: 20 video clips from Golden State Warriors games (Coach's challenge)
- **Labels**: Home team ball (7 clips) vs Away team ball (13 clips)
- **Metadata**: GSW jersey colour per game (white, blue, or black)
- **Duration**: ~1-5 seconds per clip
- **Split**: 4-fold cross-validation (15 train, 5 validation per fold)

2.2 Data Augmentation Strategy

To address limited data, employed 3x augmentation per video:

1. Standard sampling: 8 evenly-spaced frames across entire video
2. Early focus: 8 frames from first 2/3 of video (0-66%)
3. Late focus: 8 frames from last 2/3 of video (33-100%)

Additionally applied spatial augmentations: horizontal flips (50% probability) and color jittering (brightness and contrast ±20%).

Effective training set: 15 videos × 3 augmentations = 45 samples per epoch

2.3 Model Architecture

Input: Video Clip

↓

Sample 8 frames from critical region → (8, 3, 224, 224)

↓

MobileNetV2 (Pretrained on ImageNet):

- Frozen layers 1-20 (edge detection, shapes, textures)

- Trainable layers 21-30 (adapt to basketball-specific features)

↓

Per-frame features: (8, 1280)

↓

2-Layer LSTM (256 hidden units):

- Learns temporal patterns

- Models ball trajectory and player movements over time

↓

Final temporal features: (256)

↓

Jersey Colour Metadata: [1,0,0] for white → (16)

↓

Concatenate: (256 + 16 = 272)

↓

Fully Connected Classifier

272 → 128 → 64 → 2 [Away probability, Home probability]

Design Rationale:

- **MobileNetV2**: Efficient CNN (14M params) suitable for limited compute, pretrained on ImageNet for visual understanding
- **Frozen early layers**: Leverage learned edge/shape detectors, reduce overfitting
- **LSTM**: Captures temporal dependencies (e.g., "ball moving toward white jersey in frames 6-8")
- **Jersey colour embedding**: Explicit metadata helps model learn "if GSW (blue) touched last → away ball"

2.4 Training Details

- **Optimizer**: Adam with learning rate 0.001
- **Loss**: Cross-entropy
- **Regularization**: Dropout (20-40%), early stopping
- **Epochs**: 5 per fold (tested 7 epochs but led to overfitting)
- **Batch size**: 4
- **Hardware**: Google Colab (NVIDIA T4 GPU)

---

3. Results

3.1 Key Observations

- Reaching accuracy variating from 70%-90% with **80% mean accuracy**
- High variance between folds (60-100%) due to small validation sets (5 videos each)
- Model performs better on "Away" predictions
- Number of folds achieved perfect accuracy in different runs, demonstrating the model can solve the task when validation set contains "easier" examples

3.2 Error Analysis

The model struggles more with home team predictions, potentially because:

- Fewer training examples (7 home vs 13 away)
- Home scenarios may involve more complex player interactions
- Class imbalance in training data

---

4. Discussion

4.1 Achievement

80% accuracy with only 20 videos demonstrates:

- Transfer learning's power (ImageNet → basketball)
- Effective data augmentation strategy
- LSTM's ability to learn temporal patterns from limited sequences
- Value of domain knowledge (jersey colour metadata)

4.2 Lessons Learned

Initial Approach - BLIP-2 Vision-Language Model:

- Attempted using BLIP-2 with natural language prompts
- Model predicted "away" for all 20 videos (0% useful signal)
- Issue: Complex prompts caused the model to repeat questions rather than answer
- **Lesson**: Not all foundation models generalize to specific domains without fine-tuning

Pivoted to Custom Architecture:

- Built task-specific spatiotemporal model
- Leveraged transfer learning strategically (freeze early layers)
- Incorporated domain knowledge (jersey colours)
- Result: Successful 85% accuracy

4.3 Limitations

- **Small dataset**: 20 clips limits generalization to unseen game scenarios
- **Single team**: Only GSW games; may not generalize to other teams
- **Lighting variations**: All clips from similar broadcast conditions
- **No real-time inference**: Current model too slow for live gameplay

---

5. Future Work

- Expand dataset to 100+ clips from multiple teams
- Address class imbalance (collect more "home" examples)
- Experiment with attention mechanisms to focus on ball location
- Real-time inference optimization
- Integration with actual referee review systems