Basketball Data Extraction from Broadcast-Angle Video Footage

Alexander Bendeck and Niyaz Nurbhasha
Mentors: Alex Volfovsky and Harsh Parikh

Background

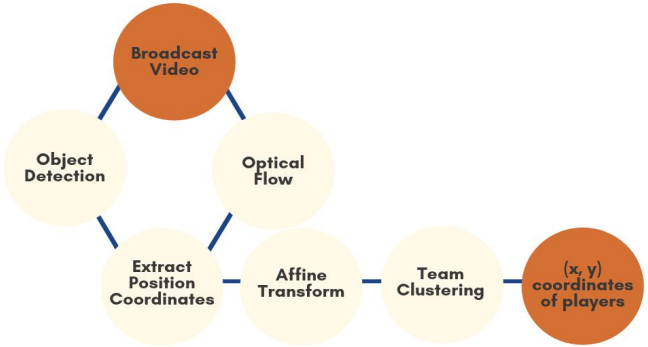
- One of four projects in a “Basketball Analytics Pipeline”
- Our goal: Extract player and ball movement data from free broadcast-angle video footage

Methods

- Instance Segmentation
  - Implemented using the Mask R-CNN neural network model
  - For computing, used an online environment called Supervisely
- Human Keypoint Detection
  - Implemented using the OpenPose API developed at Carnegie Mellon
  - For computing, used GPU nodes on the Duke Compute Cluster

Project Scope

- What we did
  - Detect players, referees, court/paint
  - Clean raw coordinate and image data
- Future work
  - Detect basketball
  - Refine algorithms
  - Expand training dataset



Object Detection

- OpenPose
  - Detects 25 keypoints on the human body
    - Often picks up noise (fans)
    - Occasionally fails to detect some players
- Mask R-CNN
  - Used weights file pre-trained on COCO dataset
  - Mostly able to recognize people on the court
    - Can struggle with overlapped players
  - Also trained on Referees, Ball, Court, and Paint
    - Paint and court were detected easily, but referees and ball were not

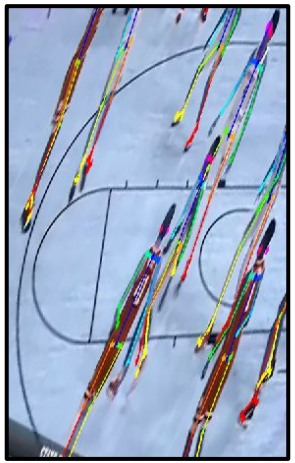
Model	Precision	Recall	F1 Score
5-epochs-referees	49.9%	68.9%	57.9%
10-epochs-referees	50.2%	63.9%	56.2%
5-epochs-ball	13.1%	12.5%	12.8%
10-epochs-ball	16.8%	31.5%	21.9%

OpenPose player detection

Mask R-CNN player detection

Data Cleaning & Processing

- Method for detecting referees without Mask R-CNN
  - Sharpen images w/ custom filters
  - Hough Line Transform to detect vertical lines on referee shirts
  - Results: On small dataset (30 players, 7 referees), all refs identified correctly
- Affine Transformation: Linear mapping between images preserving parallel lines
  - Map player locations down to an absolute court location, clean noise with court cutoffs



Above: Before Affine Transformation
Right: After Affine Transformation

- Team clustering (classifies 29/30 players correctly)
  - K-Means clustering (k = 3) to identify top pixel colors within each image
  - K-Means clustering again (k = 2) to group by most prevalent color vector
- Optical Flow: "Track" players between frames
  - Custom algorithm: Given a keypoint location in one frame, predict that the closest keypoint in the next frame is the same person
    - Good for short intervals (< 25 frames)
  - OpenCV algorithm: Follows points given by OpenPose across frames
    - Accurate only for a few frames (< 10)
    - Bad when players cross each other

Output of OpenCV Optical Flow algorithm