Cricket Ball Detection and Tracking

Name: Prithvi Dutta
College: IIT BHU
CPI: 9.07
Resume: Prihvi Dutta Resume Final.pdf

Project Details

Github: Repo Link
YouTube: Video Link

Table Of Contents

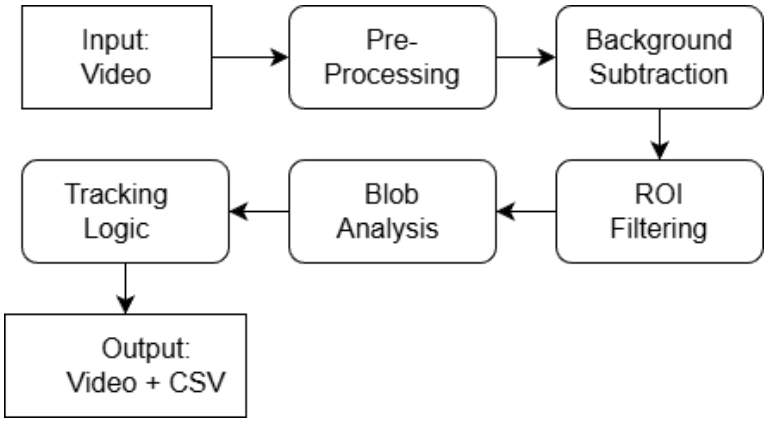
Executive Summary

The project implements a motion and shape-based detection pipeline designed to identify and track a cricket ball in broadcast video. The solution utilises classical Computer Vision techniques in OpenCV, prioritising simplicity and efficiency over the complexity of deep learning models. It's a lightweight detector that runs in real-time.

The system processes each frame through a background subtraction and blob detection pipeline. Every moving pixel is identified, followed by morphological filtering to reduce noise. A region-of-interest is applied to restrict detection outside the pitch, eliminating irrelevant background movement. Candidate objects are then filtered by geometric properties, such as area, aspect ratio, and circularity, to identify blobs that most accurately resemble a cricket ball.

The approach demonstrates a lightweight parameter-tuned solution to the problem. It performs robustly under stable lighting. As expected with motion-based segmentation, detections reduce in frames where the ball overlaps with the batsman and both objects merge into one blob. This behaviour highlights a typical occlusion challenge in classical vision pipelines and provides a clear direction for future improvement through tracking and object-level separation.

Implementation Details



Input Video: The pipeline takes a cricket video recorded from a single fixed camera as input. The system supports multiple test videos, and the paths_for() function dynamically selects the corresponding input, output, and annotation file paths. Frames from the input video are read sequentially and passed through the detection and tracking stages for processing.

```python
def paths_for(x: int):
    if x == 1:
        input_path = rf"{DATA_ROOT}\data\1.mp4"
    else:
        input_path = rf"{DATA_ROOT}\data\{x}.mov"
    output_path = rf"{DATA_ROOT}\results\output_fast_blobs{x}.mp4"
    csv_path = rf"{DATA_ROOT}\annotations\ball_annotations{x}.csv"
    return input_path, output_path, csv_path
```

Pre-Processing

Each frame is resized and normalised to reduce computation. Basic smoothing and thresholding steps are applied to reduce noise and make motion regions easier to isolate during segmentation.

```python
small = cv2.resize(frame, (0, 0), fx=SCALE, fy=SCALE)
```

```
        mask = fg.apply(small)
        _, mask = cv2.threshold(mask, 180, 255, cv2.THRESH_BINARY)
        mask = cv2.medianBlur(mask, 3)
        mask = cv2.dilate(mask, None, iterations=1)

        mask = apply_corridor_mask(mask, SCALE, W, ROI_LEFT, ROI_RIGHT)
```

Background subtraction

A background subtraction model (MOG2) is used to separate moving pixels from the static field. This produces a binary motion mask that highlights the regions where new motion appears, forming the basis for candidate-ball selection.

```
def create_background_subtractor(history, var_threshold, detect_shadows):
    return cv2.createBackgroundSubtractorMOG2(
        history=history,
        varThreshold=var_threshold,
        detectShadows=detect_shadows
    )
```

ROI Filtering

A fixed-pitch corridor region of interest is applied. Movement outside the corridor is not considered to improve precision.

```
def apply_corridor_mask(mask, scale, w, roi_left, roi_right):
    rx1 = int(w * roi_left * scale)
    rx2 = int(w * roi_right * scale)
    mask[:, :rx1] = 0
    mask[:, rx2:] = 0
    return mask
```

Blob Analysis

Connected components are extracted from the filtered motion mask and evaluated based on their geometry. They are filtered using area, aspect ratio and circularity to retain shapes most consistent with a ball.

Tracking Logic

Each frame where a valid blob is detected, the centroid is computed and appended to the trajectory. A visibility flag is maintained for missing frames.

```python
def detect_round_blobs(mask, scale, min_area, max_area):
    contours, _ = cv2.findContours(
        mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    results = []

    for c in contours:
        area = cv2.contourArea(c)
        if area < min_area or area > max_area:
            continue

        peri = cv2.arcLength(c, True)
        if peri == 0:
            continue

        circularity = 4 * math.pi * area / (peri * peri)
        if circularity < 0.80:
            continue

        c = (c / scale).astype(int)
        x, y, w, h = cv2.boundingRect(c)
        aspr = w / float(h)
        if aspr < 0.8 or aspr > 1.2:
            continue

        cx = x + w // 2
```

```
        cy = y + h // 2

        results.append((c, x, y, w, h, cx, cy, area, circularity))

    return results
```

Output

The final results include a processed video with the detected centroid, along with a per-frame annotation file containing the frame index, centroid coordinates, and visibility status.

```
csv_writer.writerow([frame_id, cx_frame, cy_frame, visible])
        out.write(vis)
```

Performance

We introduced an optional optimisation to reduce computation by skipping odd-numbered frames.

```
if SKIP_FRAMES and (frame_id % 2 == 1):
            frame_id += 1
            continue
```

The Model does a good job of predicting the ball when its background is static. When the ball is in front of the batsman, the blob of the batsman and the ball merge, and the model fails to detect the ball. The same can be seen in this image

Also, ball-like objects like the logo on the player's chest sometimes get detected, leading to false positives, as can be seen in the image below.

The Model detects the ball first based on motion and then applies geometric constraints, such as area, circularity, and aspect ratio, to find a suitable candidate.

YOLO Attempt

After the Classical CV approach, we tried to train a YOLO model. Annotated 331 images using Roboflow and trained the model. The model did not perform as expected. It hallucinated frequently and misidentified objects as balls. These can be seen in the images attached.

It detects the cap as a ball. One workaround would be if the video were extremely high quality, I believe it would give better results.

Results


Brave File Edit View History Bookmarks Profiles Tab Window Help
Fri 10 Oct 5:48 PM
THE BALL PARK
LIVE STREAMING
GC
Aditya Jha * 2 (4)
Balwinder Singh 6 (4)
GCR 8-0 / 1.2 (20) CBN
PROJECTED SCORE: 120
1st Inning
Mayank Tomar 0-5 (0.2)
cricheroes


NEXGEN CORPORATE LEAGUE SEASON 11 | TELANGANA SUPER KINGS VS ALL STARS
All stars (Ysportz) 3-0
1.0
CRR: 3.00 PROJECTED SCORE: 60
vs Telangana Super ...
JEETU 2(2) RAVI CHANDRA 1(4) NAVEEN 0-0(0)
OV.
cricheroes


11-09-2025 Sun 15:32:14
SANJAY
69
7EVENPEGS 141-6 14.3 CRR: 9.72 PROJECTED SCORE: 194 vs Uttarakhand Stars
VIVEK KUMAR 11(4)* RACHIT 17(12) KESHAV 2-30(2.3) OV. 4 2 1
cricheroes


NEXGEN CHAMPIONS TROPHY SEASON-4 | Mavericks VS Forties Eleven
UMPIRE
19
Mavericks 13-0 2.2 CRR: 5.57 PROJECTED SCORE: 111 vs FORTIES ELEVEN
SAI RAM 4(6)* SUSHANTH KODELA 8(8) JAI KISHAN 0-5(1.2) OV. 0 1
cricheroes