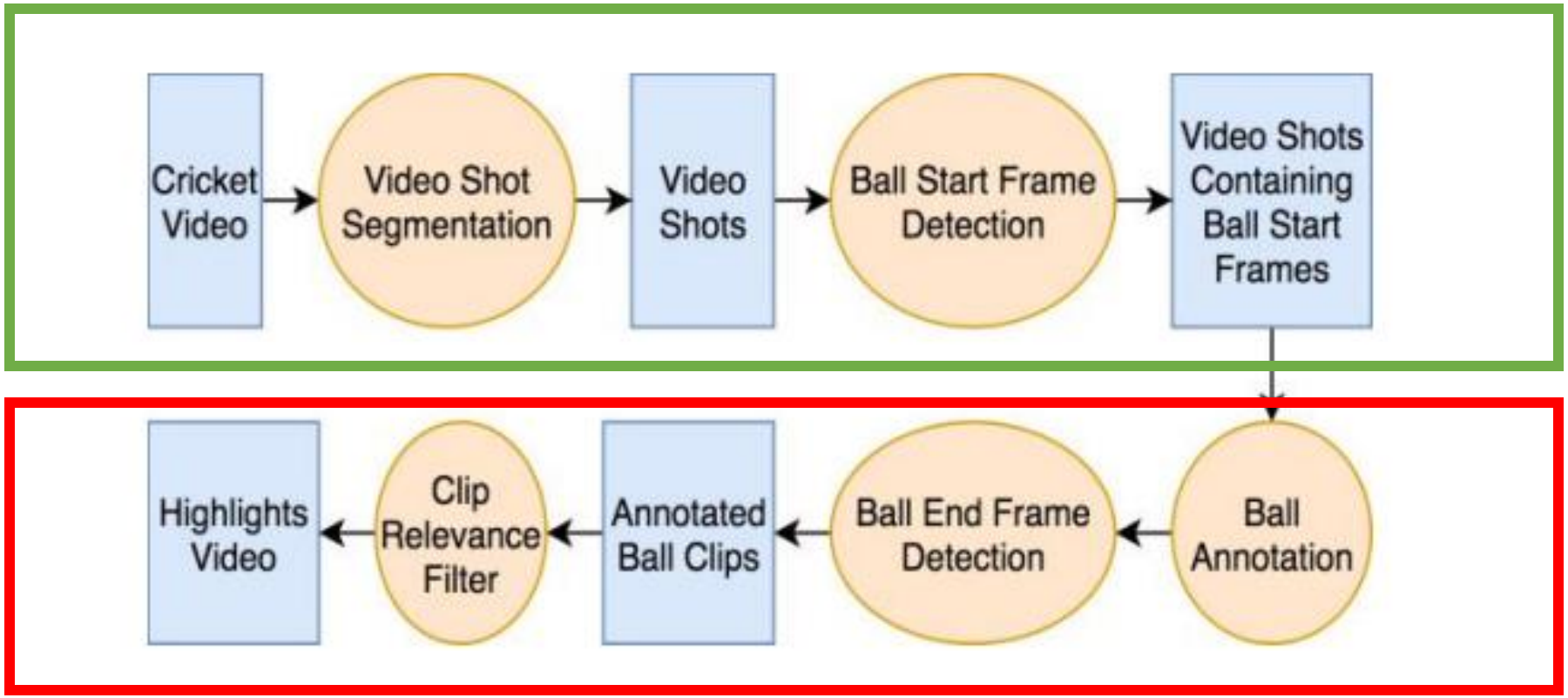
Automatic Highlights Generation for Cricket Matches

Alina Sarwar

Motivation

- Cricket game is longer than most other games
- Manual generation of highlights will be very time-consuming
- For this project, we work with the shortest cricket format
  i.e. T20

Proposed Methodology

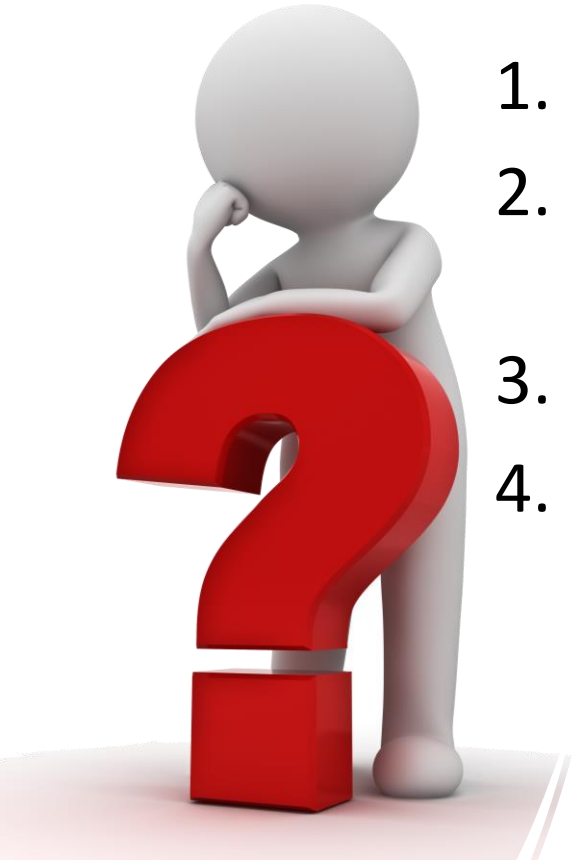


Agarwal et al, "Automatic Annotation of Events and Highlights Generation of Cricket Match Videos", in International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.

P1 Results: Some ball start frames from a match

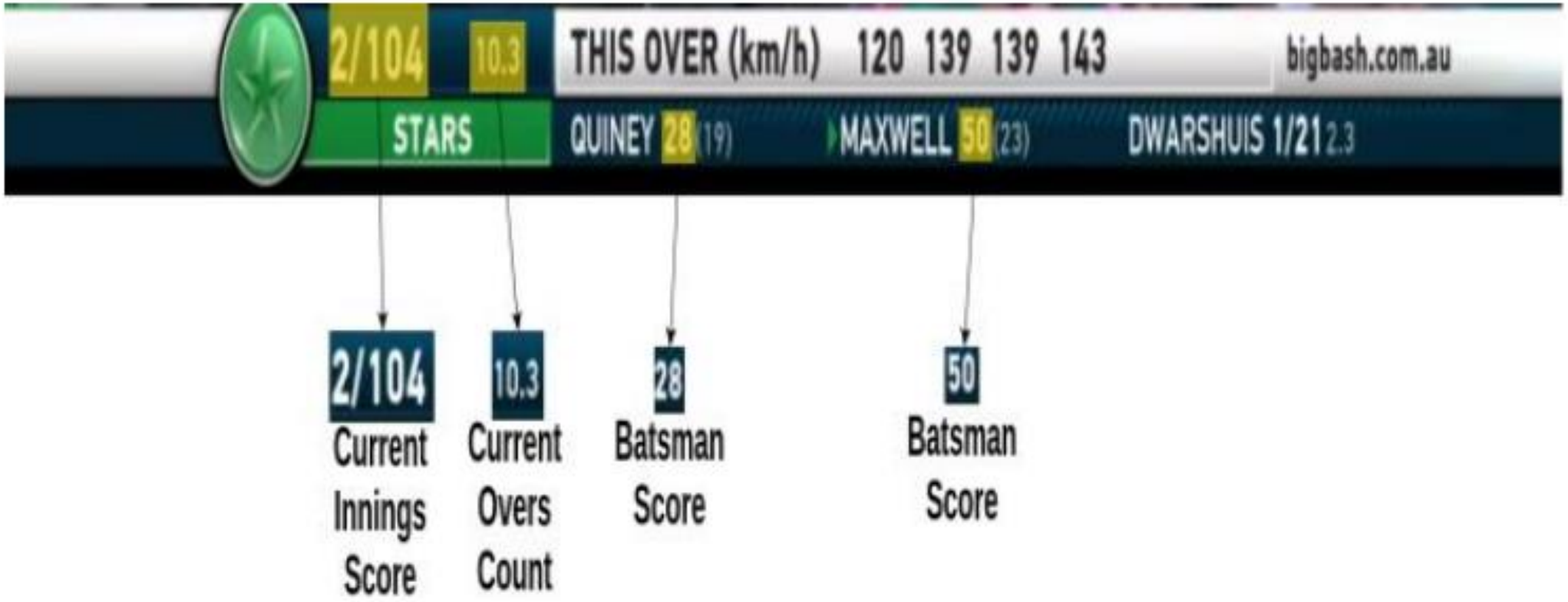
Goal for P2: Track game progress using OCR

Optical Character Recognition (OCR) is the process that converts an image text into machine-readable text format.



1. Ball start frame --> CNN classifier
2. Ball end frame --> Run OCR on subsequent frames until over count changes
3. Annotate ball clip --> compare this ball's score with the next
4. Filter relevant clips

Regions of Interest in a Scorebox



Broadcasters use different scorebox designs

- ROI marked manually before processing a video
- Remain the same for the entire video

Preparation of input for OCR

飞桨 PaddleOCR

original

enlarged

sharpened

black and white

https://github.com/PaddlePaddle/PaddleOCR

original frame

cropped scorebox

OCR inputs

OCR outputs

'12.5' '1/89' 'WHITE 37' 'FINCH 47'

Ball Clip Inclusion Criteria

1. Four or more runs were scored on a ball
2. The wicket falls on a ball
3. The ball is the last of the innings

All such ball clips are then stitched together to generate the final highlights clip using the Python library MoviePy.

Model performance on 4 innings from 2 matches

Innings	ACTUAL INNINGS					OUR HIGHLIGHTS					
	Length (mins)	Sixes	Fours	Wickets	Last ball	Length (mins)	Sixes	Fours	Wickets	Last ball	False Positives
1	104	1	8	8	1	7	1	7	8	1	1
2	69	2	12	2	1	6	2	10	1	1	4
3	87	3	16	6	1	8	3	15	6	1	3
4	89	1	7	8	1	6	1	5	7	1	3

Innings	Precision	Recall	Processing time (mins)
1	0.89	0.94	13
2	0.78	0.82	9
3	0.89	0.96	10
4	0.82	0.82	10
AVERAGE	0.85	0.89	10.5

Balls that do not satisfy the inclusion criteria

Loopholes in the proposed framework

- Robustness - model not immune to disappearance of the score box
- Not able to identify ball end frame in case of wide or no ball
- Processing cost too high for longer format matches

Track score count changes for ball end frame detection

- Helpful in case of wide/no ball when only score count changes
- Sometimes score changes in increments e.g. `1--> 2--> 4`
- Can lead to incomplete ball clips and undercounting.

- Use both over and score count, with preference to the former.

Future Work

- Evaluate model on more matches
- Make model robust to score box disappearance
- Include a ball in the highlights if it is a milestone (i.e. batsman scores a multiple of 50)

Questions?