👉 ***You must explain why this video is suitable or problematic.***

I have picked the sample video not only because it was provided, but also because it is a good video for analysis. It has both important aspects a side view angle and it is phone recorded. As we know, a side view is important for analysis because through a side view we can clearly analyze knee bend movements, hip movement, arm movements, body leaning forward or backward, and the movement of the front and back legs. This would not be possible from a front view because the movements would overlap and could cause mistakes in keypoint detection. The video is phone recorded which is important because it connects the analysis to the real world as coches or cricketers will not always carry a DSLR or professional camera.

The video is good because it is taken from a side view and recorded using a phone as explained above. The angles made by the knee and arm, as well as changes in hip movement, are clearly visible which makes analysis easier. However the video also has some limitations. During fast bat swing movements there is lag and blur around the wrist, leg, and elbow, which reduces keyoint detection accuracy. Real world data is always messy because it comes from real situations and if the data were perfectly clean, it would not be realistic or useful for building a practical model.

Another limitation is that only a side angle video is available, which reduces accuracy because movements cannot be analyzed from multiple angles. For example, it is difficult to judge how close or far the bat is from the camera. Also, since the video is captured from only one side, joints closer to the camera are detected more accurately than those on the far side which can introduce bias in keypoint accuracy.

My approach

I did not want to treat this as just another assignment to submit and move on from. While working on it, I kept thinking that if I get selected, this kind of work would be part of real projects, not just practice. So I tried to build something that could work in realistic conditions instead of only looking good on paper.

I spent some time looking at different pose estimation models like YOLO,RTMpose MediaPipe. I finally chose MediaPipe because it works reasonably well even when the data is limited, which is the case in an assignment, and because it handles phone recorded videos more reliably.

While working with the video, it became clear that real world data is not clean, and that is normal. The practice net in the video creates a lot of visual noise, and sometimes the model gets confused and predicts joints where they are not actually visible. Instead of ignoring this issue, I tried to handle it in a simple way by smoothing the pose data over multiple frames so that sudden false detections do not affect the final results too much.

Overall, the aim was not to build something perfect, but to understand the problems that come with real cricket videos and try to deal with them in a practical way. I wanted the final output to be something that makes sense and could be useful in a real setting, even if it is not flawless.

Why I chose MediaPipe

I did not pick a model at random for this project. I spent time looking at different pose estimation models such as YOLO and RTMPose before making a choice. I finally selected MediaPipe because it suited the type of data and constraints of this assignment better.

One important reason was familiarity. I have worked with MediaPipe before and I understand how it behaves in real situations. This helped me move faster and focus more on analyzing cricket movements instead of spending time dealing with setup issues or unexpected behavior. Since the goal was movement analysis and not model training this felt like the right choice.

Another reason was the nature of the data. Real world videos are usually messy especially in cricket practice environments where nets motion blur and partial occlusion are common. In this video the practice net blocks parts of the body many times. MediaPipe handled this situation more reliably by re detecting the player when visibility dropped which helped keep the tracking more stable.

MediaPipe also provides more body landmarks compared to many other models. Models like YOLO usually output fewer keypoints while MediaPipe gives 33 landmarks including detailed foot points. Since footwork is very important in cricket having access to ankle heel and toe information made the analysis more meaningful. RTMPose works well when the data is clean and when there is enough training or fine tuning data. In this assignment the data was limited and the video had noise and net occlusion. RTMPose usually needs more data to stay stable in such conditions. MediaPipe worked more reliably out of the box for this kind of real world video.

Because I was familiar with MediaPipe I was also able to handle AI hallucinations caused by the net. Instead of trusting every prediction the pipeline looks at multiple frames together and smooths out sudden false jumps. This helps keep the movement data steady and closer to how a human would naturally interpret the motion.

What problems did you observe

Only a side view phone video was available. Based on the analysis of this side on phone footage I observed the following main technical challenges.

Jitter Even though the player was moving smoothly the keypoints on the skeleton often appeared to shiver or jump slightly from frame to frame. This happens because the AI treats each frame as a separate image and does not always remember where the limb was a moment earlier. I handled this by applying a smoothing filter but this remains a natural challenge when working with mobile recorded footage.

Occlusion
Because the video is recorded from a side view the batsman's own body often blocks certain joints. For example during the backlift or follow through the far side arm or leg is hidden behind the torso. This forces the AI to guess the joint position which reduces accuracy for the hidden joints.

Wrongkeypoints
This was one of the most noticeable issues. The practice net creates a grid like pattern that can confuse the AI. At times the model mistakes a junction in the net for a human joint such as the elbow or wrist which causes incorrect keypoints to appear away from the actual body. As mentioned earlier if the data is not messy it is not real. I used this messy data to design a filter that ignores these unrealistic joint jumps.

Motionblur
Because the bat swing is very fast the phone camera cannot always capture it clearly. This results in blur around the wrist and bat. When the pixels become unclear the AI struggles to locate the exact joint position. This is a common limitation when working with standard 30 fps or 60 fps phone videos.

Metrics Defined

The lead elbow angle measures the angle formed by the shoulder, elbow, and wrist of the front arm. This metric is important because in cricket a high lead elbow is considered the correct technique for playing straight drives. It helps the bat travel in a more vertical path toward the ball. If the elbow angle is too tight or collapses early, the bat can slice across the ball instead of meeting it cleanly. By tracking this angle, we look for a clear extension during the backlift followed by a controlled bend during the drive.

The lead knee stability metric measures the angle formed by the hip, knee, and ankle of the front leg. This is important because a strong and stable front leg provides the base for power and balance in a cricket shot. If the knee bends too much or collapses at the moment of impact, the batsman can lose power and stability. Tracking this angle helps identify whether the player is leaning into the shot with a firm and balanced base.

Trunk lean measures the angle of the upper body formed by the shoulder and hip relative to a vertical line. This metric helps assess the player's balance during the shot. A well balanced batsman keeps their head and body aligned over the ball. Excessive backward lean can result in mistimed shots, while too much forward lean can cause loss of balance. By calculating this angle, we can understand how the player's weight shifts during the shot and whether it is controlled.

Improving the Accuracy

If I had more time, I wouldn't just use a "ready-made" AI. I would add "Human Physics" to the code. Right now, if the netting confuses the AI, the player's arm might "jump" to a weird spot. I'd teach the AI that a human arm has a fixed length and can only move so fast. If the AI tries to make the arm "teleport," the code would block it. I would also use "Memory" layers. Instead of looking at one frame at a time, the AI should remember the last few frames to "guess" where the hand is, even if the bat or the net is blocking it.

Adapting it to Cricket

To make this a real tool for coaches, I'd add "Bat Tracking." A coach doesn't just care about the player's elbows; they care about where the bat is going. I'd also build an "Auto-Cutter" so the AI automatically finds the start of the swing and the moment the ball hits the bat. This way, the coach doesn't have to watch a 10-minute video just to find one 3-second shot.

What data would I collect?

I'd go for "Messy" Data. Most AI is trained on clean videos. I would record thousands of videos specifically through nets (green, black, thick, thin). This teaches the AI that the net is just background noise. I'd also use High-Speed video (Slow-Mo) so that even when the bat is moving at 100mph, the picture stays sharp and the AI doesn't get confused by the blur.

How I would split the data

- **Train (70%)**
- **Validation (15%)**
- **Test (15%)**

Improvement Plan

If I had more time and access to more data the first and most important improvement would be collecting more cricket specific videos. This would include videos from different players skill levels and environments such as practice nets open grounds and matches. Having more variety would help the system handle different body types playing styles lighting conditions and camera placements more reliably.

I would also experiment with using more than one pose estimation model instead of relying on a single model. MediaPipe could be used as the primary model because it works well on phone recorded videos while another model like YOLO Pose or RTMPose could be tested on cleaner frames. Comparing outputs or selecting results based on confidence could improve overall accuracy.

Another improvement would be using multiple camera angles. Adding a second camera from behind the stumps along with the side view would help reduce depth related errors and bias. This would allow better understanding of bat position body rotation and joint symmetry which are difficult to judge from a single angle.

I would also improve how noise and hallucinations are handled by using more advanced temporal filtering and phase based analysis. Instead of smoothing the entire video equally different smoothing strategies could be applied during fast movements like the swing and lighter smoothing during stable phases like the stance.

Finally I would evaluate improvement by checking keypoint stability reduction in jitter and consistency of joint angles across similar movements. Visual inspection of the skeleton overlay and comparison of metrics before and after improvements would help confirm whether the system is becoming more reliable and useful for real cricket coaching.