Cricket Shot Detection Using Computer Vision

Sreya Sirivella Vamsikrishna Jayaraman
University at Buffalo
sreyasir@buffalo.edu, vjayaram@buffalo.edu

August 17, 2024

1 Project Overview

This project focuses on developing a system for cricket shot type prediction using pose estimation techniques. The key body points are identified during the execution of a shot and classified it into categories such as cover, sweep, pull, flick, straight, square cut. Then the key points are processed for neural network models for classification of cricket shots. Then real-time prediction of the shot is done. This project offers valuable feedback on batting techniques, aiding coaches and players in skill improvement. Additionally, it has the potential to automate the commentary process in cricket.

2 Approach

Pose Estimation: Detectron2's pre-trained *keypoint_rcnn_R_101_FPN_3x* model is used to identify keypoints on the cricketer's body from the extracted images. These keypoints serve as essential features for the subsequent classification task.

Shot Classification: Two custom neural networks, 'Model' and 'ModelBN', are implemented using the extracted keypoints to classify the cricket shots into predefined categories. 'Model' uses a standard fully connected layer architecture with ReLU activations and dropout layers, while 'ModelBN' incorporates batch normalization layers for improved stability and convergence.

3 Experimental Protocol

The dataset was created by extracting frames from cricket match videos at the moment of bat-ball contact, identified using spectrogram analysis of the audio. The images were labeled by shot type (cover, sweep, pull, flick, straight, square cut), and data augmentation was applied to enhance the dataset.

Evaluation Metrics: The success of the model was evaluated using a combination of metrics, including accuracy to measure the proportion of correctly classified shots, and precision and recall to assess the model's ability to make correct classifications and capture all relevant instances. The F1-score was used to quantitatively evaluate the balance between precision and recall, ensuring a comprehensive measure of the model's performance.

Compute Resources: We used MAC M2 pro device cpu for training th model. Python, PyTorch, Detectron2, and TensorFlow/Keras were used in a conda environment to manage dependencies.

Implementation: We implemented a script that extracted audio from video files and performed spectrogram analysis to detect the precise moment of bat-ball. This was useful to create a dataset. Then we used the pre-trained Detectron2 model from the model zoo was used for accurate keypoint detection of cricketers. The neural network models used for shot classification, including 'Model' and 'ModelBN', were designed and implemented independently, with 'ModelBN' incorporating batch normalization layers to enhance performance. Additionally, the real-time prediction system, capable of processing video streams and predicting shot types on-the-fly, was developed entirely from scratch.

Model Architecture: The architecture comprises four linear layers, with the first three layers acting as hidden layers, each containing 2048 units. ReLU activation functions are applied after each linear transformation to introduce non-linearity, followed by dropout layers with a 25 percent dropout rate to reduce the risk of overfitting. The input layer accepts a feature vector of size 34, and the final output layer produces a 6-dimensional vector, corresponding to the six classes in the classification task.

Model BN Architecture: The model is a fully connected neural network that incorporates Batch Normalization and Dropout for improved training stability and regularization. It consists of two hidden layers, each with 2048 units, followed by ReLU activation functions. Batch Normalization is applied after each hidden layer to normalize activations, reducing internal covariate shifts and accelerating training. Dropout, with a 30 percentage rate, is included to prevent overfitting.

Figure 1: Keypoint detection and shot prediction

4 Results

- **Key Points Estimation**: The key point estimation results are shown in fig 1.
- **Model Accuracy**: The highest accuracy is achieved by Model with 64 percent testing accuracy and the results for training and validation accuracies and losses over epochs are shown from fig 2 to 5. Model has precision and recall of 62 percent, and ModelBN has precision and recall of around 60.
- **Real-Time Prediction**: The prediction for straight drive is shown in fig 6.

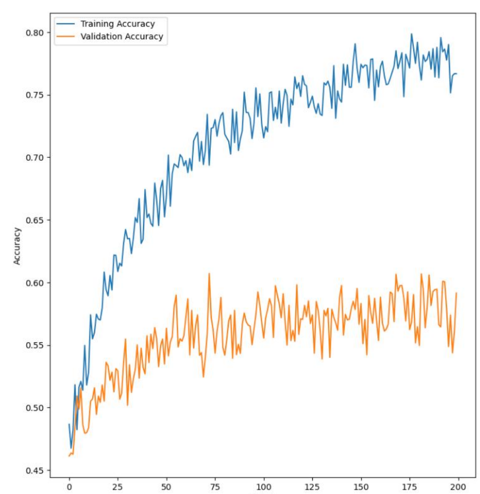

Figure 2: Accuracy over epochs for Model

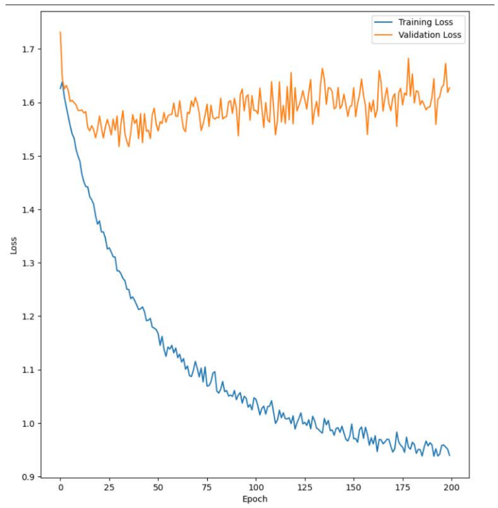

Figure 3: Loss over epochs for Model

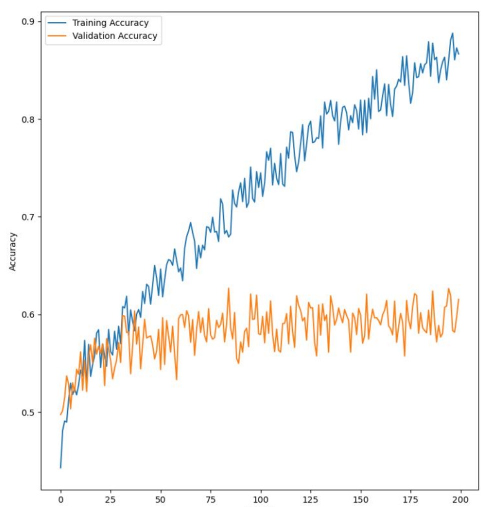

Figure 4: Accuracy over epochs for ModelBN

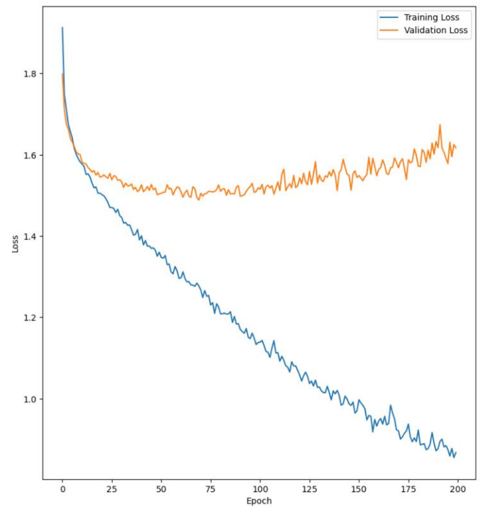

Figure 5: Loss over epochs for ModelBN

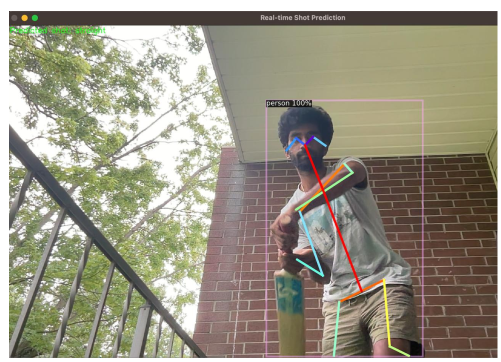

Figure 6: Real-time prediction results

5 Analysis

Limitations: Provided the best performance with accurate key point detection and classification for high quality images and showed reduced accuracy when images were of low quality due to less reliable key point detection.

Advantages: **Model**, a fully connected neural network with dropout, provides a robust approach to managing high-dimensional input data, with dropout layers effectively reducing overfitting. **ModelBN**, is similar to Model but builds on this by incorporating Batch Normalization, which normalizes layer activations to improve training stability and accelerate convergence, making the model more efficient and resilient during training. **Detectron algorithm**, used for pose estimation, offer high precision in detecting and segmenting objects by utilizing region proposal networks and multi-task learning.

6 Discussion and Lessons Learned

This project provided us significant insights into the practical challenges of implementing pose estimation and model classification. We learnt the importance of data quality, The quality of image directly impacted the performance of pose estimation and shot classification. High-resolution, well-lit videos significantly enhance model accuracy, underscoring the need for high-quality data. We can use GAN to improve the quality of images, specifically using DeblurGAN.

References

[1] S. Venugopalan, S. An, S. Bhat, and D. Parikh, *Activity Recognition for Sport Videos using Deep Convolutional Networks*, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 530-536. doi:10.1109/CVPRW.2015.7301349.

[2] R. Wadhwa and D. Rajesh, *Cricket Shot Classification using Machine Learning and Image Processing*, *International Journal of Computer Applications*, vol. 174, no. 9, pp. 10-15, 2021. doi:10.5120/ijca2021921407.

[3] L. Agnihotri, A. Mittal, and D. Jain, *Cricket Shot Prediction using Computer Vision and Deep Learning*, In *Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 1238-1242. doi:10.1109/ICIP42928.2021.9506355.

[4] S. Pund, S. Phansopkar, and S. Apte, *Automated System for Cricket Shot Classification using Neural Networks*, *International Journal of Computer Science and Information Technologies*, vol. 9, no. 3, pp. 135-139, 2018.

[5] A. Singh and M. Sharma, *Cricket Shot Classification and Analysis using Deep Learning*, *Journal of Sports Analytics*, vol. 6, no. 2, pp. 79-88, 2020. doi:10.3233/JSA-200383.