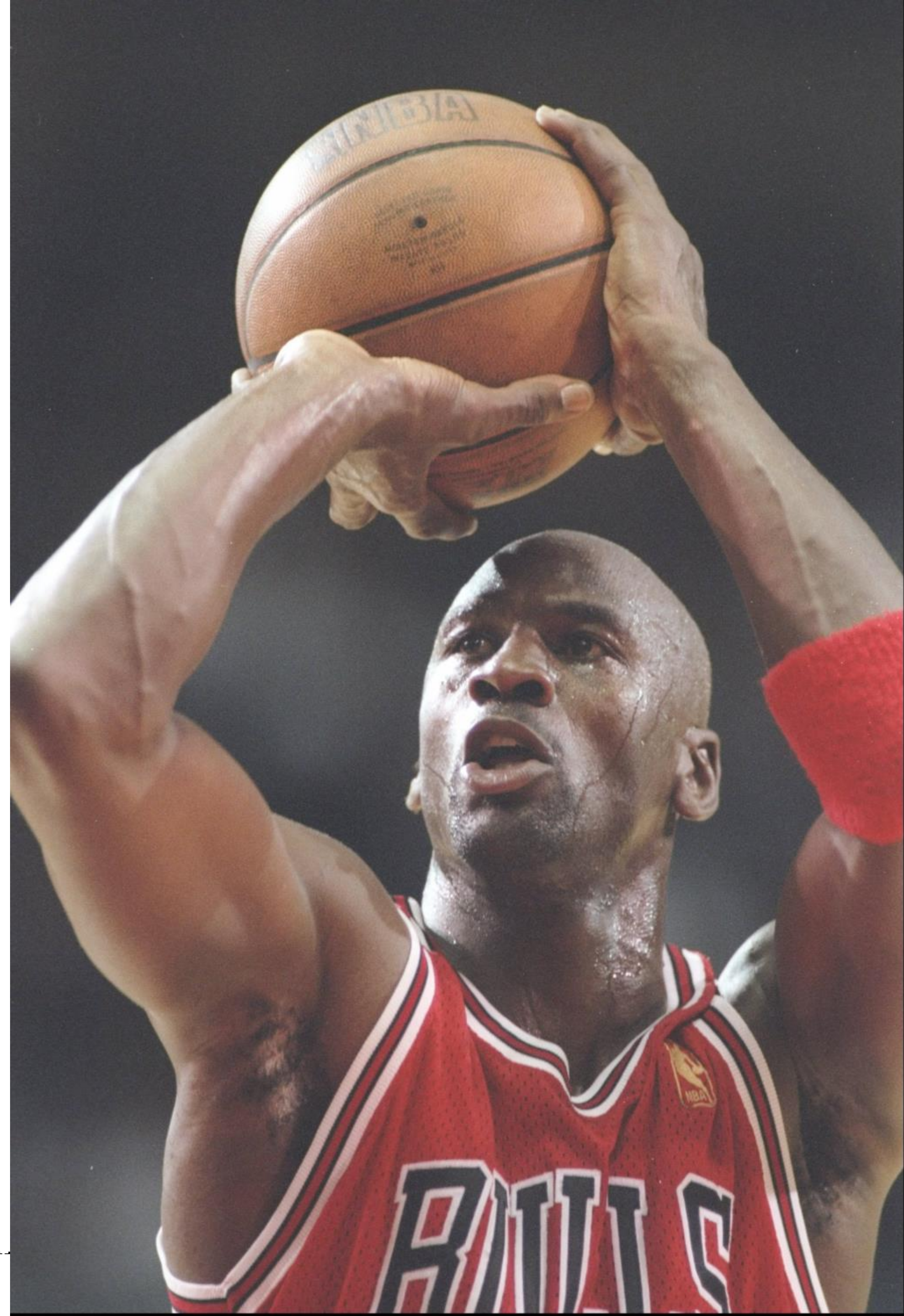
CV based shooting pose classifier

Problem statement

- **Basketball players both amateur and experienced often have biomechanical inconsistencies which lead to no replicable shooting form and inconsistent results.**



Application:

- Coaches' assistant
- Facilitates individualized training for specific players
- Personalized coaching approach
- Streamlines team selection trials for new players

Impact:

- Team shooting improvement leading to better team performance
- Prevent injuries.

Literature survey

- Jeffries, Colby T., "Sports Analytics With Computer Vision" (2018). Senior Independent Study Theses. Paper 8103
- Basketball Video Analysis. (n.d.). Pose Estimator. Retrieved from [URL]: https://github.com/stephanj/basketballVideoAnalysis/wiki#pose-estimator
- Duane Knudson (1993) Biomechanics of the Basketball Jump Shot—Six Key Teaching Points, Journal of Physical Education, Recreation & Dance, 64:2, 67-73, DOI: 10.1080/07303084.1993.10606710
- Fan, J., Bi, S., Wang, G., Zhang, L., & Sun, S. (2021). Sensor Fusion Basketball Shooting Posture Recognition System Based on CNN. In B. Gao (Ed.), Journal of Sensors (Vol. 2021, pp. 1–16). Hindawi Limited. https://doi.org/10.1155/2021/6664776

INSIGHTS

- Wearable sensors have been employed to analyze both a player's body posture and shooting technique.
- In our approach, we exclusively utilize computer vision and machine learning models to categorize and recommend personalized recommendations for refining shooting styles.
- Previous models only account for shot quality and can indicate that there is a problem with either the mechanics or the trajectory or both but aren't specific in diagnosing the problem
- There are multiple shot trackers, but they are more intended to maintain a scoring estimate and track shots taken not analyze biomechanics

Dataset and Preprocessing

Normalization: The dataset is normalized to get the output irrespective of the height of the player. (Frame averaging, Temporal Normalization)

Media pipe: MediaPipe Pose Detection is a computer vision library that accurately locates and tracks human body key points in images or video. Using deep learning, it identifies key body joints, enabling applications like gesture recognition and posture analysis.

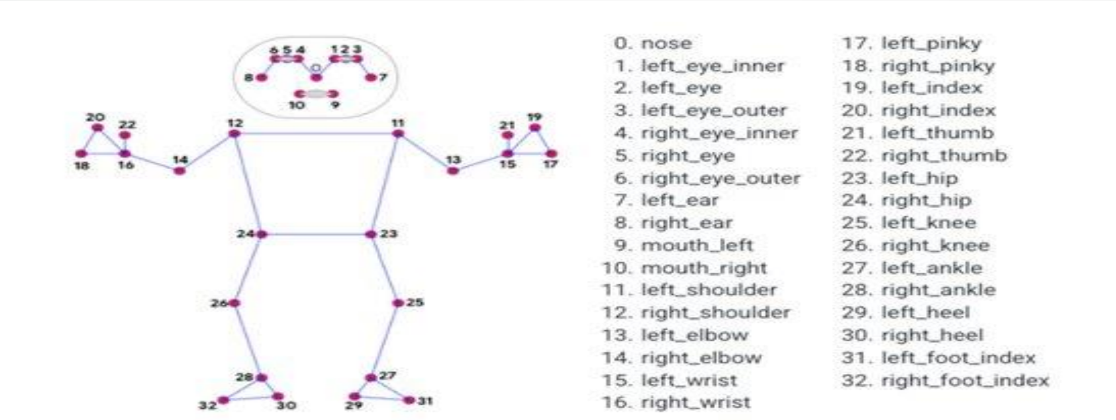


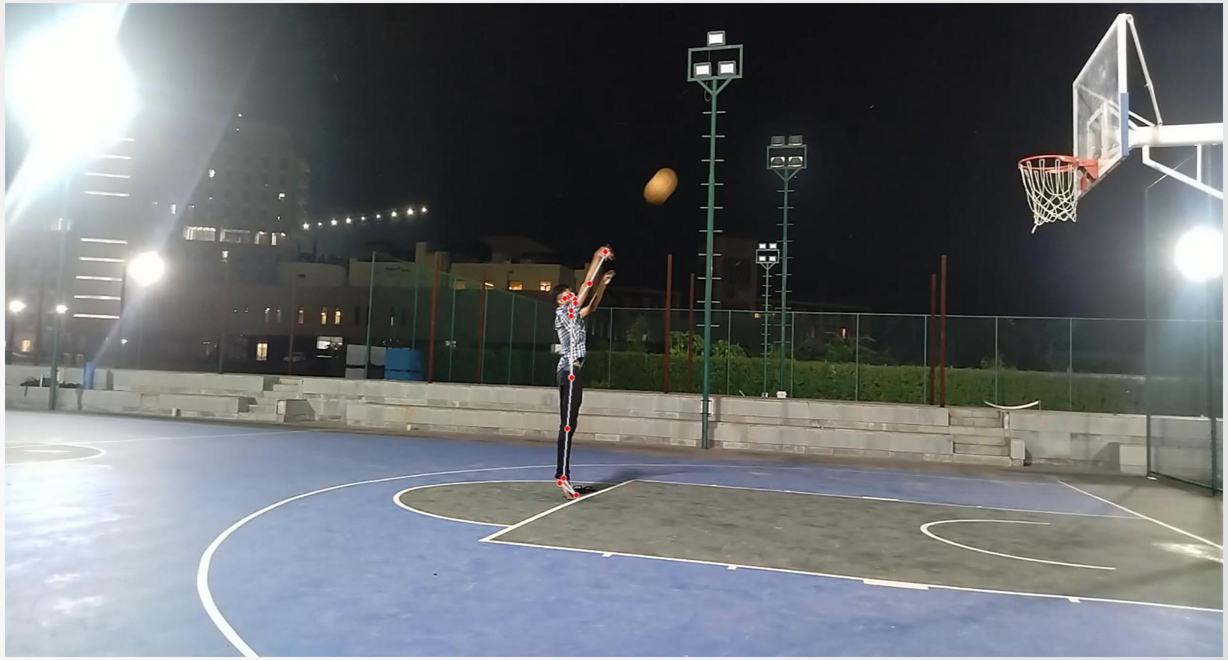
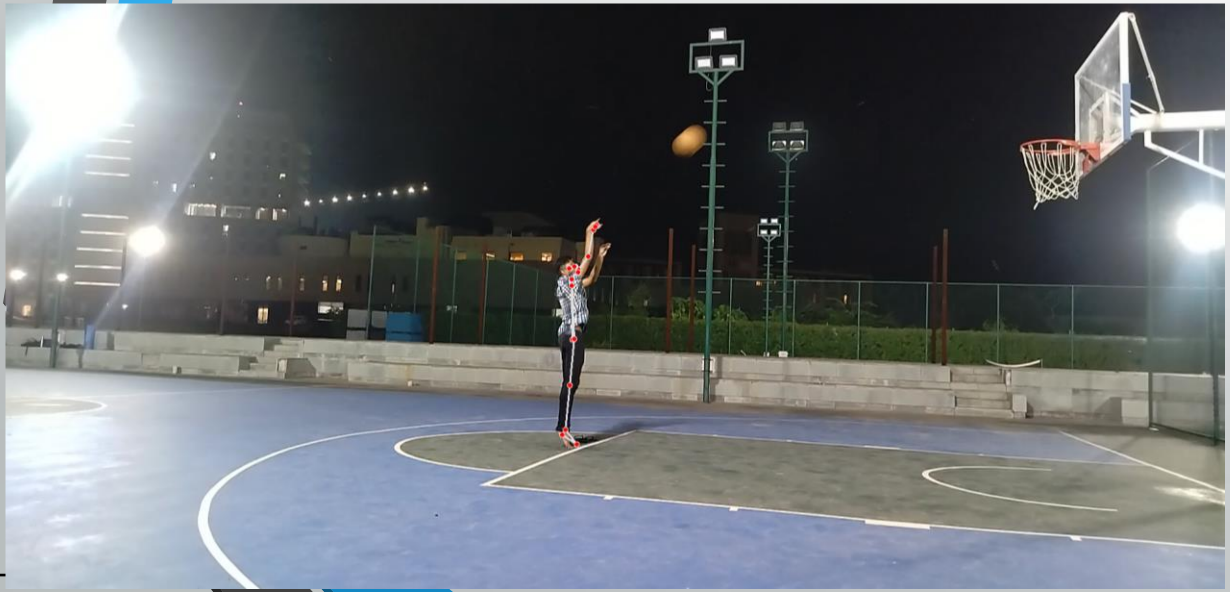
Figure 1. Definition of landmarks in MediaPipe Pose [22].

Data engineering

- We carried data augmentation using frame transformations in order to increase the size of the available dataset.
- We also filled out the feature values for missing frames using averaging of previous and next frames or replacement with previous frame.
- We then converted the dataset into a npy file set for all the videos(frames) and then concatenated them with separators
- Following this we used a 0.2 test to 0.8 train split of our dataset in order to create the dataset that would be used to train most of the models.

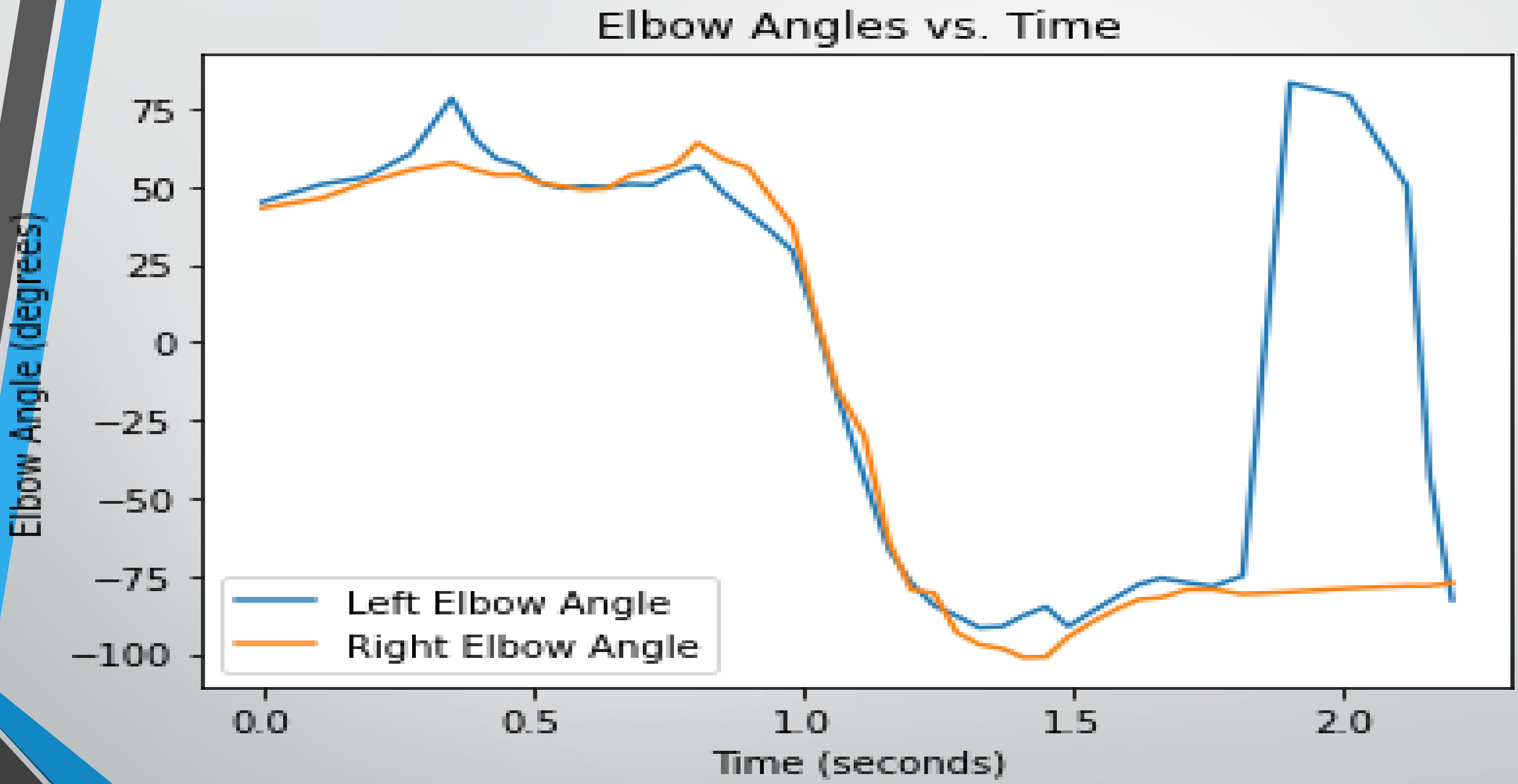



POSE ESTIMATION:


KEY Features

- Shooting Hand Angle: Calculate the angle formed by the shooting hand and the shooting arm. A consistent angle may indicate good form.
- Release Point: Determine the height of the release point when the ball is released. A consistent release point is essential for accuracy.
- Alignment: Check the alignment of the shoulders, hips, and feet during different phases of the shot. Misalignment can affect shooting accuracy..
- Balance: Assess the player's balance during the shot. Imbalances can result in missed shots.
- Elbow Position: Check the position of the shooting elbow during the shot. A consistent elbow position can indicate proper shooting mechanics.


Elbow Angles vs. Time
Elbow Angle (degrees)
75
50
25
0
−25
−50
−75
−100
0.0
0.5
1.0
1.5
2.0
Time (seconds)
Left Elbow Angle
Right Elbow Angle

Features reduction

- So far, we haven't encountered a requirement to reduce the dimensionality of our features as biomechanics studies, have found that all the key features are equally important to the shooting motion.
- Without understanding component-wise contribution PCA would only give us an idea of shot quality which has already been studied but not contributing factors which can help us build a robust classifier.

Methodology 1(SVM)

- Support Vector Machines (SVM) classify data by finding a hyperplane that maximally separates different classes in a high-dimensional space. It selects the optimal hyperplane by maximizing the margin, the distance between the hyperplane and the nearest data points (support vectors). SVM is effective for both linear and non-linear classification tasks using kernel functions.
- **Ability to Handle High-Dimensional Data:**
  - **Explanation:** SVM draws a line (hyperplane) to separate different types of basketball shooting forms. Even if we have lots of details (high-dimensional data) from pose detection, SVM is great at figuring out the important stuff.
- **Robust to Overfitting:**
  - **Explanation:** SVM has a way of not getting too caught up in the details during training. It strikes a balance between learning from the examples it sees (training) and being smart enough to handle new, unseen examples (testing).
- **Suitable for Small to Medium-Sized Datasets:**
  - **Explanation:** SVM is like a superhero for smaller datasets. Even if we don't have tons of videos of basketball shots, SVM can still make sense of the information we have and give us reliable results.

Methodology 2 (Random Forest)

- Random Forest is an ensemble learning algorithm that builds multiple decision trees during training. It randomly selects subsets of features and data, allowing each tree to learn independently. The final prediction is a combination of individual tree predictions, often resulting in improved accuracy and robustness. Random Forest is effective for classification and regression tasks in diverse datasets.

- **Effective in Separating Classes:**
  - **Explanation:** Random Forest builds multiple decision trees, each contributing to the final classification through majority voting. This ensemble approach ensures robust decision-making, crucial for distinguishing various qualities in basketball shooting forms.
- **Versatility with Decision Trees:**
  - **Explanation:** Comprising diverse decision trees, Random Forest adapts to complex relationships within biomechanical data. By using various splitting criteria, it accommodates both linear and non-linear patterns, enhancing its flexibility in capturing nuanced shooting techniques.
- **Robust to Overfitting:**
  - **Explanation:** Random Forest mitigates overfitting by aggregating predictions from multiple trees. This prevents the model from becoming overly specific to training examples and ensures reliable generalization, particularly important in real-world scenarios with variable shooting forms.

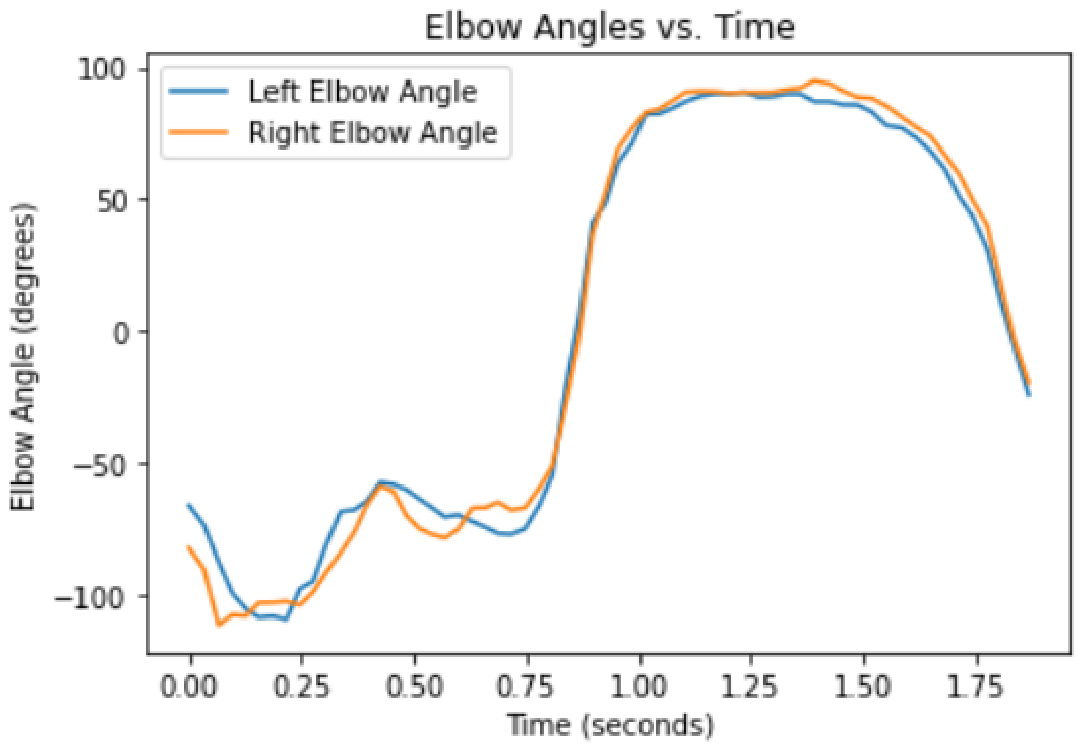
Methodology 3(k-means clustering)

- K-means is a clustering algorithm that partitions data into K clusters by iteratively assigning points to the nearest cluster center and updating the center based on the mean of assigned points. The process repeats until convergence, optimizing cluster assignments and centroids. K-means is widely used for unsupervised learning, identifying patterns in data.

- **Efficient Grouping of Similar Shooting Forms:**
  - **Explanation:** K-means efficiently categorizes shooting forms by iteratively assigning forms to clusters based on shared features. This aids in identifying groups of similar biomechanical patterns within the dataset.
- **Unsupervised Learning for Pattern Discovery:**
  - **Explanation:** K-means, being an unsupervised learning algorithm, uncovers inherent patterns in the data without the need for predefined categories. This is particularly valuable in discovering natural groupings and trends in basketball shooting forms.
- **Scalability and Computational Efficiency:**
  - **Explanation:** K-means is computationally efficient and scalable, making it well-suited for large datasets. It quickly converges to find cluster centroids, allowing for the analysis of numerous biomechanical features in a computationally effective manner.

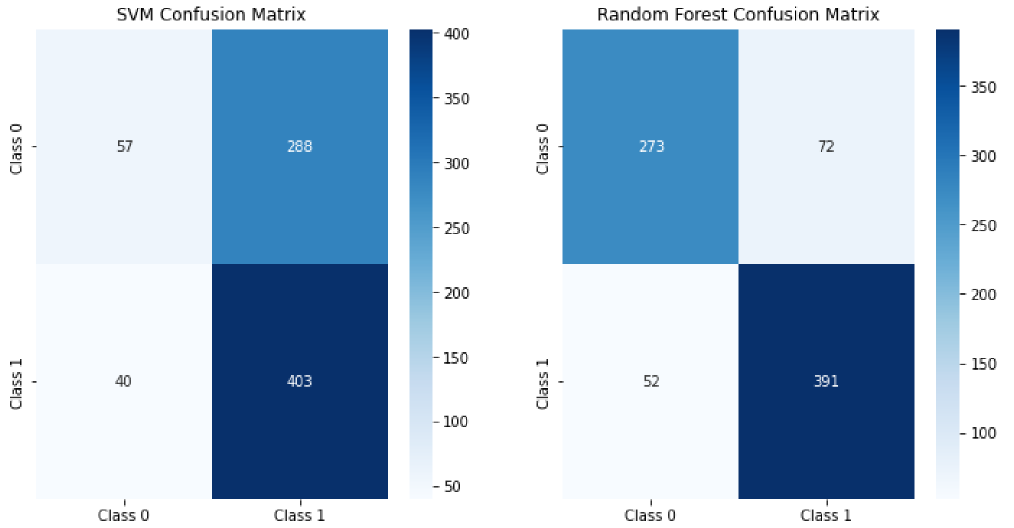
Challenges

- Data collection was an issue as finding shooter with good form and consistency is not very common amongst amateur shooters
- Identifying good shots in the dataset had to be done manually thus leaving the possibility of some human error
- Some of the classifiers showed significant change in behaviour upon increasing dataset size and after augmentation


Elbow Angles vs. Time
Left Elbow Angle
Right Elbow Angle
Elbow Angle (degrees)
Time (seconds)
100
50
0
−50
−100
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75

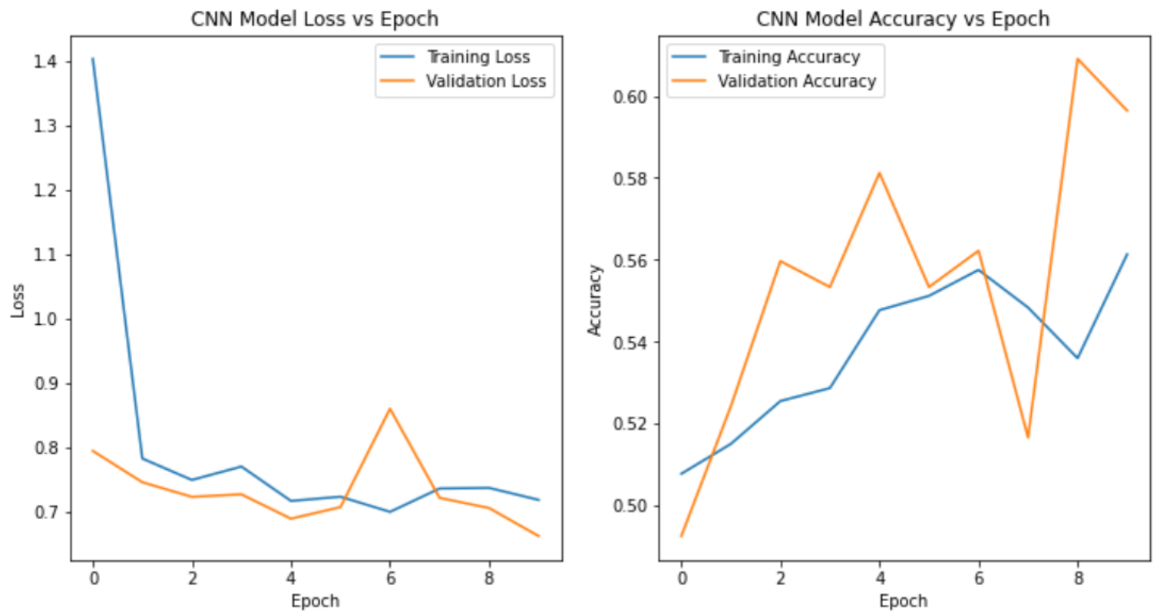
Results

Results of SVM and Random forest (1)



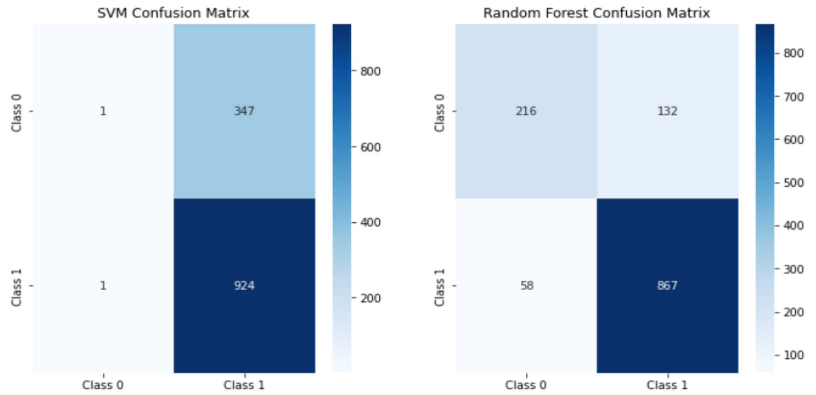

```
SVM Accuracy: 0.583756345177665
SVM Precision: 0.5832127351664255
SVM F1 Score: 0.710758377425044

Random Forest Accuracy: 0.8426395939086294
Random Forest Precision: 0.8444924406047516
Random Forest F1 Score: 0.8631346578366446
```

CNN model loss and model accuracy



SVM and Random Forest (2)

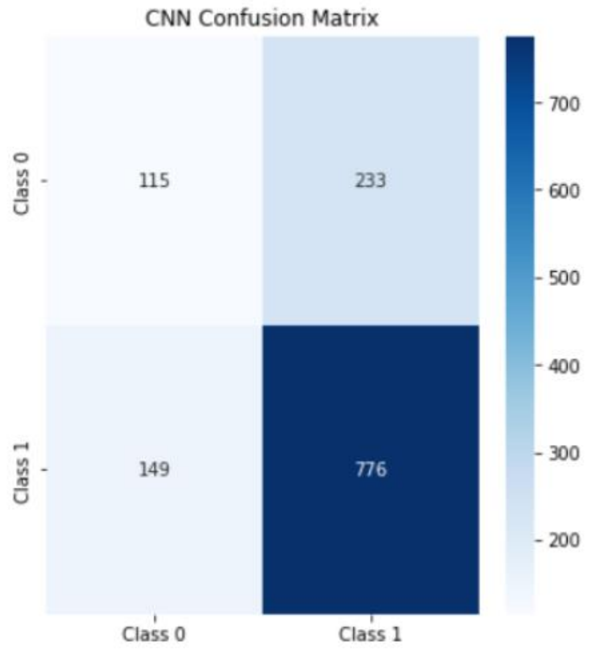
```
Processing video: C:/Users/Dell/Documents/BSE/sem 5/MLPR_PRCT/MLPR_dhruv\VID_20231130_204925.mp4
Processing video: C:/Users/Dell/Documents/BSE/sem 5/MLPR_PRCT/MLPR_dhruv\VID_20231130_204934.mp4
```

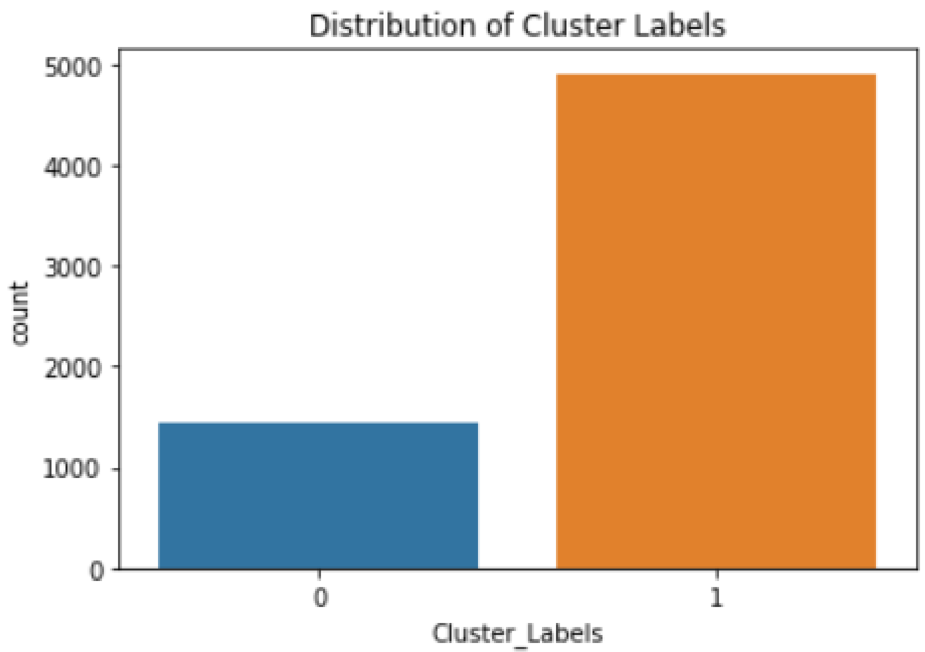



```
SVM Accuracy: 0.7266300078554595
SVM Precision: 0.7269866247049567
SVM F1 Score: 0.8415300546448086

Random Forest Accuracy: 0.8507462686567164
Random Forest Precision: 0.8678678678678678
Random Forest F1 Score: 0.9012474012474011
Epoch 1/10
159/159 [==============================] - 1s 2ms/step - loss: 0.9556 - accuracy: 0.6486 - val_loss: 0.6794 - val_accurac
y: 0.6465
Epoch 2/10
```

CNN and RNN

```
CNN Accuracy: 0.6999214454045561
CNN Precision: 0.7690782953419227
CNN F1 Score: 0.8024819027921407

RNN Accuracy: 0.7266300078554595
RNN Precision: 0.7266300078554595
RNN F1 Score: 0.8416742493175614
```

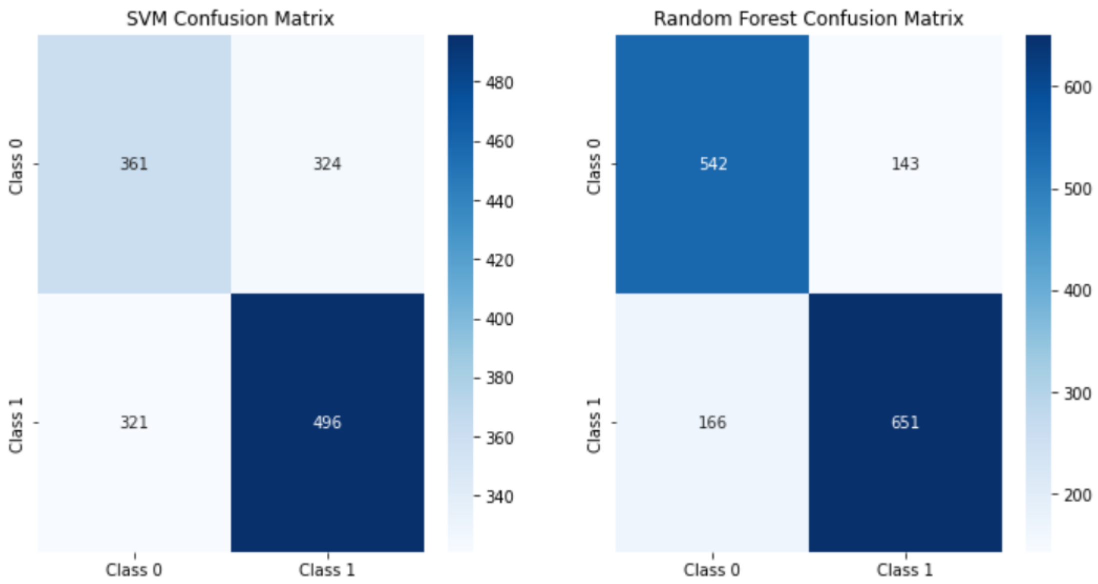





```
Count of data points in each cluster:
1    4907
0    1454
Name: Cluster_Labels, dtype: int64
```

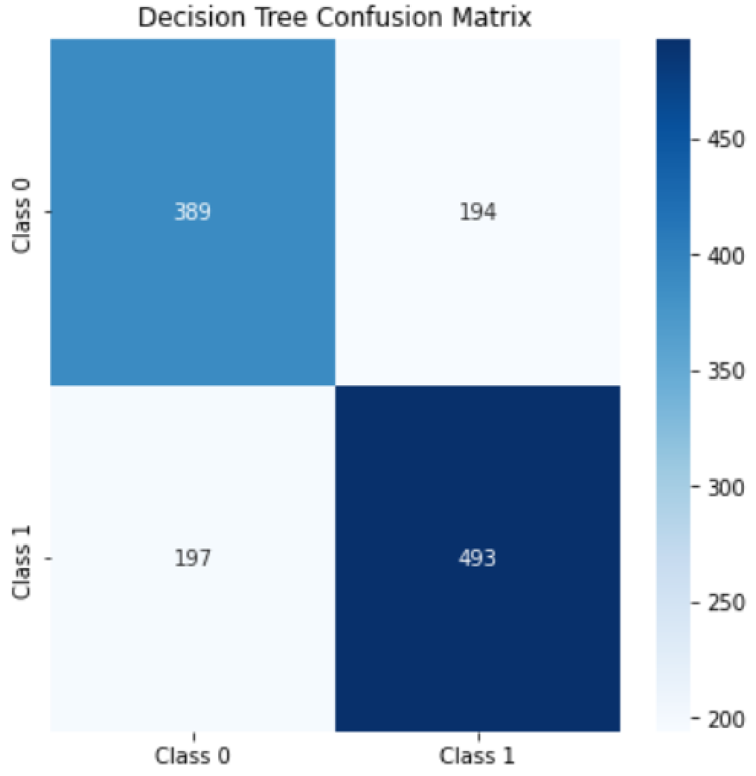
Clustering

SVM and Random Forest (3)

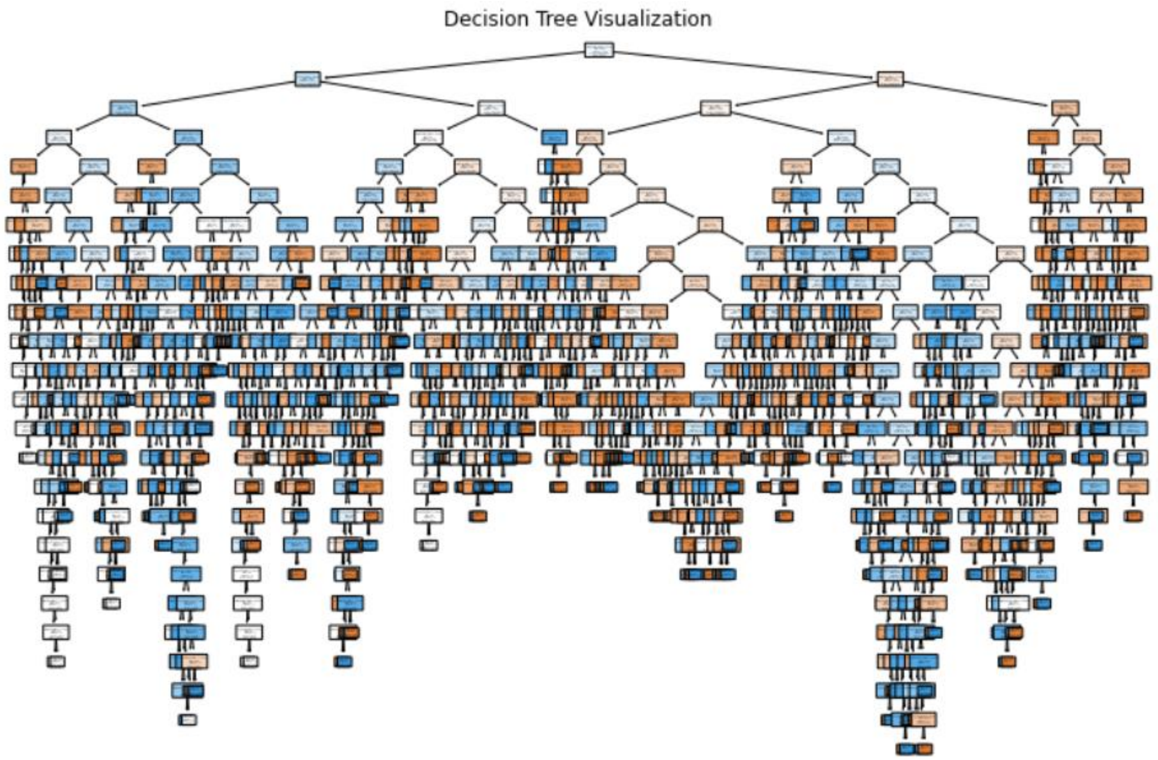



```
SVM Accuracy: 0.570572569906791
SVM Precision: 0.6048780487804878
SVM F1 Score: 0.6059865607819181

Random Forest Accuracy: 0.7942743009320905
Random Forest Precision: 0.8198992443324937
Random Forest F1 Score: 0.808193668528864
```

```
Decision Tree Accuracy: 0.6928515318146111
Decision Tree Precision: 0.7176128093158661
Decision Tree F1 Score: 0.7160493827160495
```



Decision tree matrix



Visualization of Decision tree

Deployability

- The model is almost deployable for the use case envisioned with a clear understanding of good vs bad shooting based on both replicability and comparison
- An increased sampling dataset when integrated with team practices would improv model accuracy and provide good 2-way feedback
- Possibility of pruning and quantising the models in some instances to be able to deploy as real time edge device based coaching assistants to the Plaksha basketball coach.

Future

We have built a combination of consistency and comparison metrics that the models look at. Based on this we will offer recommendations to improve upon and follow up as a feedback loop.

Deployment on an edge-based device like RPi   for ML could allow for it to become a virtualized basketball coaches assistant