Weightlifting Form Analyser Using Neural Networks Final Project Report

Ayush Choudhary[1],Sidharth Chopra[2],Akshay Shingade[3]

University of Colorado Denver[1], University of Colorado Denver[2],University of Colorado Denver[3]

Abstract

The project has been created with the purpose of implementing Computer Vision techniques to various weightlifting movements in order to study the form of these movements and then provide feedback on the movement that can be used by the user to optimize the trajectory of the motion which in turn will enhance the performance as well as reduce the chances of injuries. In our approach we've used traditional computer vision algorithms/techniques and have combined that with basic geometry and neural networks to get our results.

1. Introduction

In today's fast-paced world, weightlifting has gained immense popularity as people find it challenging to prioritize fitness amidst their busy schedules. As a result, resistance training has emerged as a preferred mode of fitness. Not only casual fitness enthusiasts, but also a significant number of athletes are increasingly adopting weightlifting as a way to train. This trend is on a steady rise with each passing day.

Weightlifting, powerlifting, and crossfit are highly technical sports that demand precision and leave very little room for error. As a result, they are prone to a high incidence of injuries. Poor form while executing a particular movement is a leading cause of injury, with injuries to the deltoids, pectorals, lower back, knees, elbows, and hips being common during compound movements. Additionally, weak muscle groups or muscular imbalances can also contribute to injury risk. However, these injuries can be prevented by lifting weights with proper form and posture. Individuals who have undergone surgeries in the past may have a higher risk of injury in specific body parts.

By carefully tracking exercise form, lifters can gain valuable insights into the different phases of a lift, enabling them to enhance their performance and reduce their risk of injury. Such applications can prove to be a valuable asset for coaches, as they can monitor the progress and health of their athletes. Furthermore, these tools can also benefit athletes without access to professional coaches and novices seeking to learn proper form and technique.

2. Review of Literature

2.1 Methodology

After conducting an extensive study of various papers related to our project's sub-parts, we narrowed our focus to four specific research papers. These papers delved into topics such as body key-points, posture detection, sports motion recognition, barbell trajectory measurement using smartphones, barbell kinematics, and real-time repetition counting. These concepts have all influenced our project in some manner. One paper that we examined explored computer vision research in sports as a whole, providing us with valuable insights into how computer vision algorithms can be practically applied in real-world scenarios. To ensure that we obtained the most relevant information, we limited our search to papers published after 2017 and utilized multiple computer science research paper repositories with relevant keywords.

2.2 Main Findings

In our search for studies on pose estimation, we have come across existing literature that primarily pertains to the accurate tracking of exercise form, repetition counting, and models capable of distinguishing between correct and incorrect form. Furthermore, various studies have demonstrated the reliability of smartphone-based applications for measuring these statistics.

Computer vision technology is extensively used in sports, with several well-established applications and ongoing research topics (Thomas et al., 2017). Examples include multi-camera ball and player tracking, as well as vision-based tracking systems that enable image overlays. This technology has been leveraged to enhance refereeing and generate valuable statistics for coaches, ultimately enhancing the user experience. However, one challenge currently faced by this field is the lack of high-quality datasets for certain movements. This paper has provided us with an understanding of the current research topics within this field and how we can apply our own ideas to the realm of weightlifting to extract valuable statistics.

According to Balsalobre-Fernandez et al. (2020), Computer Vision-based smartphone applications can yield reliable and valid measurements for various parameters of barbell kinematics in complex Olympic weightlifting movements such as Snatch. In comparison to advanced motion capture systems, these simpler applications are cost-effective and require minimal equipment and markings, making them less time-consuming to use. Additionally, they do not require calibration and only need a tripod for stability. The user-friendly interface and high level of accuracy are notable benefits of these applications. However, it should be noted that research on movements that involve rotation, such as shot put and discus throw, is limited in this area. Through this paper, we have gained insight into the analysis of intricate barbell movements and confirmed the effectiveness of smartphone hardware for conducting such studies.

In this paragraph, the study of sports key posture detection and motion recognition using deep learning techniques is discussed. The study was done in early 2022 by Shaohong Pan. The paper focuses on using CNNs to extract and classify features from video frames to obtain key frames and using time series information to make predictions about the movement of the specimen based on its position data. The paper provided insights into using relevant frames from the video to minimize computation time, which could be used to determine the angles between joints during different phases to calculate certain values that will be used in the exercise_Doc component. The paper mainly dealt with 2D pose estimation to 3D pose prediction, and the challenge with it was self-occlusion. The future work for this study is based on improving the performance and occlusion issues of the framework.

The paragraph discusses a study (Alatiah & Chen, 2020) that uses neural networks for exercise recognition and counting of repetitions. The study compared the performance of three different methods, including the proposed method, the smartwatch, and the GymCam. The neural network was trained on the UCF101 dataset, which contained push-ups, pull-ups, and squats. The study generated body key points for each exercise, with 7437 push-ups, 12707 pull-ups, and 14357 squats body key points. The smartwatch had a higher accuracy rate (99.96%) than the proposed method (98.4%) in exercise recognition, while the proposed method was slightly better in counting repetitions. The study relates to the project as it also uses key points to calculate the angle between different joints and provide feedback on optimizing form. However, a major downside of the study was its inability to differentiate between competitors and judges, which is currently being worked on.

3. Implementation discussion and dataset creation

In order to create a dataset for our project, we recorded videos of ourselves performing bicep curls with both perfect and incorrect form. We took care to record from three different angles to ensure comprehensive coverage: from the right, from the left, and from a general perspective. Using the powerful and user-friendly mediapipe library for pose estimation, we were able to calculate the angles created at the elbow for each frame of each video. These values were then stored in a csv file for easy access and manipulation. To facilitate our analysis, we labeled the correct exercises as 1 and the incorrect exercises as 0, thus creating a clear and intuitive dataset. The python scripts pose_1.py, pose_2.py, pose_general_bicep.py and pose_left_angle_bicep.py do the task of creating the csv files.

```python
def findAngleSP(self,frames):
    h,w,c=frames.shape
    self.results=self.pose.process(frames)
    lm= self.results.pose_landmarks.landmark

    l_shldr_x = int(lm[11].x * w)
    l_shldr_y = int(lm[11].y * h)

    l_elbow_x = int(lm[13].x * w)
    l_elbow_y = int(lm[13].y * h)

    l_wrist_x = int(lm[15].x * w)
    l_wrist_y = int(lm[15].y * h)

    r_mouth_r_x = int(lm[9].x * w)
    r_mouth_r_y = int(lm[9].y * w)

    r_hip_r_x = int(lm[23].x * w)
    r_hip_r_y = int(lm[23].y * w)

    b_angle=np.arctan2(r_mouth_r_y-r_hip_r_y, r_mouth_r_x-r_hip_r_x)
    b_angle = np.abs(b_angle*180.0/np.pi)

    l_angle= np.arctan2(l_elbow_y-l_shldr_y, l_elbow_x-l_shldr_x)-np.arctan2(l_elbow_y-l_wrist_y, l_elbow_x-l_wrist_x)
    l_angle = np.abs(l_angle*180.0/np.pi)

    if l_angle>180.0:
        l_angle=360-l_angle

    return l_angle, b_angle
```

Fig 1- The code above is calculating the angle that is being created at the left bicep and the angle that is being created by the back and then is returning that angle. This angle is stored onto the respective csv files.

Once we created our dataset, the next step was to train our neural network to classify the bicep curl exercises as either correct or incorrect. We decided to train separate models for each of the three camera angles we had recorded: the right side view, left side view, and the general view. This allowed us to capture any variation in the exercise form across the different angles. We split our dataset into training and validation sets to train and evaluate our models respectively.The scripts extract_csvdata_bicep_left.py, extract_csvdata_bicep_misc.py and extract_csvdata_bicep_right.py do the task of normalizing our data, splitting them into training and testing datasets which we use for training the neural network.

```
left_bicep_angles_data_correct = pd.read_csv('/content/drive/MyDrive/bicep_left_angle.csv')
left_bicep_angles_data_incorrect = pd.read_csv('/content/drive/MyDrive/left_wrong.csv')

# Extract the correct body angles and labels from the data
elbow_angles_correct = left_bicep_angles_data_correct.iloc[:, 0].values.reshape(-1, 1)
back_angles_correct = left_bicep_angles_data_correct.iloc[:, 1].values.reshape(-1, 1)

# Extract the incorrect body angles and labels from the data
elbow_angles_incorrect = left_bicep_angles_data_incorrect.iloc[:, 0].values.reshape(-1, 1)
back_angles_incorrect = left_bicep_angles_data_incorrect.iloc[:, 1].values.reshape(-1, 1)

# Combine the two columns into a single array
angles_correct = np.concatenate((elbow_angles_correct, back_angles_correct), axis=1)
angles_incorrect = np.concatenate((elbow_angles_incorrect, back_angles_incorrect), axis=1)

# Set all labels to 1 since the angles are correct
labels_correct = np.ones(len(angles_correct))

# Set labels to 0 for incorrect angles
labels_incorrect = np.zeros(len(angles_incorrect))

# Concatenate the correct and incorrect body angles into a single array
angles_all = np.concatenate((angles_correct, angles_incorrect), axis=0)

# Create labels for all angles
labels_all = np.concatenate((labels_correct,labels_incorrect), axis=0)

# Combine the angles and labels into a single DataFrame
left_bicep_angles_data = pd.DataFrame(np.concatenate((angles_all, labels_all.reshape(-1, 1)), axis=1), columns=['elbow_angle', 'back_angle', 'label'])

# Split the data into training, validation, and test sets
train_angles_left, test_angles_left, train_labels_left, test_labels_left = train_test_split(angles_all, labels_all, test_size=0.2, random_state=42)

train_angles_left, val_angles_left, train_labels_left, val_labels_left = train_test_split(train_angles_left, train_labels_left, test_size=0.2, random_state=42)
```

Fig 2- The code above is preprocessing the data and then splitting it for training and testing.

To train our models, we used the popular Python library Keras with a Tensorflow backend. Our dataset consisted of the angle measurements at the elbow joint from the three camera angles for each repetition of the exercise, along with a binary label indicating whether the repetition was performed correctly or not.

We experimented with different neural network architectures and hyperparameters, and used cross-validation to select the best model for each camera angle. Once we had trained our models, we saved them using the Python library pickle. Saving the trained models in this way allowed us to reuse them later without having to train them again from scratch. The python scripts train_model_left.py, train_model_misc and train_model_right.py do the task of training the neural network.

```python
from keras.models import Sequential
from keras.layers import Dense,Dropout
from keras.regularizers import l2

# Define the neural network architecture
model = Sequential()
model.add(Dense(32, activation='relu', kernel_regularizer=l2(0.01), input_dim=2))
model.add(Dense(16, activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dense(8, activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model on the training data
history =model.fit(train_angles_left, train_labels_left, validation_data=(val_angles_left, val_labels_left), epochs=20, batch_size=10)

# Evaluate the model on the test data
loss, accuracy = model.evaluate(test_angles_left, test_labels_left)
print('Test accuracy:', accuracy)

model.save('bicep_left.h5')
```

Fig 3- The code above is showing the architecture of our neural network and then saving it.

Overall, creating and training the neural network was a challenging task, but we were able to successfully create models that could classify the bicep curl exercises based on the angle measurements. With the trained models saved, we could now integrate them into our application to provide real-time feedback to the user on the correctness of their exercise form.

4. Results

Below are the screenshots from our outputs.

We have received a testing accuracy of 79% for our model that is trained on the bicep-curl data that has been collected from the right side angles while the model which has been trained on the bicep curl data for the left side angles has achieved a testing accuracy of 76%. We are sharing the graphs comparing the training and validation accuracy of our models.

It can be clearly seen that the accuracy of our model is improving as the epochs are increasing, showing a good learning curve and at the same time, we can see that the validation accuracy is somewhat better than the testing accuracy which shows that our data is not overfitting/underfitting and is providing expected results.

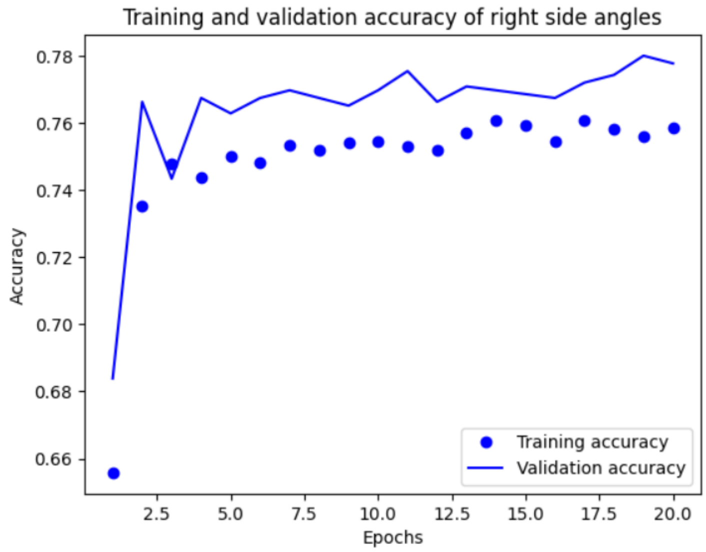

Fig 4- The graph above compares the training and validation accuracy of our bicep-curl model from the right-side view.

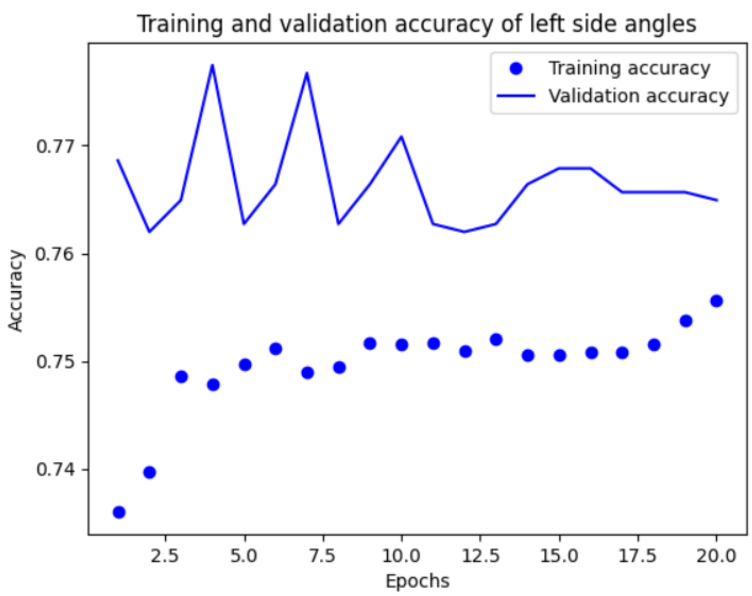

Fig 5- The graph above compares the training and validation accuracy of our bicep-curl model from the left-side view.

5.Future Work

As we look to the future, we envision expanding our project to encompass a broader range of lifts and exercises beyond the bicep-curl. While our current focus has been on perfecting our model for this specific exercise, we are eager to apply our approach to other compound movements. To achieve this goal, we plan to develop additional models that will be trained to recognize and evaluate proper form for each lift.

One of the key ways we hope to improve our model's accuracy is by introducing multi-angle inputs. By allowing users to upload videos of their lifts from different perspectives, we can provide our model with more data to analyze and enhance its ability to identify correct form. With multiple angles, our model can take into account different aspects of the lift that may not be visible from a single perspective. We plan to merge the data obtained from these different inputs to produce a more comprehensive output for the user.

6.Uses/Application

The purpose of this project is to develop a tool that can be utilized by various professionals such as sports scientists, performance coaches, strength and conditioning coaches, and trainers. The application aims to provide these professionals with comprehensive insights into the biomechanics of their athletes by offering precise readings. This would enable coaches to create personalized training plans to address the weak points of their athletes. It is important to note that while this tool will not replace human coaches, it will certainly make their work much easier. Additionally, this application can prove to be extremely beneficial for athletes from financially struggling nations.

In addition to catering to the needs of professionals, this application can also be utilized by beginners and intermediate lifters who wish to monitor their form while performing complex compound movements. This tool can be particularly useful for them as it can help prevent injuries while simultaneously enhancing their performance.

7. Conclusion

Computer vision techniques have revolutionized the field of sports by providing unique insights and introducing new methods for analyzing sports movements. Recent studies have shown that the applications of computer vision are not just limited to ball-tracking and player tracking, but they can be used for refereeing, injury prevention, and performance optimization. Our project builds upon this body of work and takes it a step further by analyzing weightlifting movements in depth and extracting new statistics that would be of great interest to coaches, performance scientists, athletes, and hobbyists alike.

To achieve this goal, we will be training our model on multiple videos, through which we will extract data such as the athlete's posture, speed in a particular phase of a lift, and angles of the joints during different phases. Our model will then provide the users with comparisons and improvement tips based on this data. By doing so, we aim to provide a tool that would not only help coaches and athletes to monitor and optimize their performance but also help beginners and intermediate lifters prevent injuries and improve their form.

8. References

1) Computer vision for sports: Current applications and research topics by Graham Thomas, Rikke Gade,Thomas B.Moeslund,Peter Carr,Adrian Hilton. https://doi.org/10.1016/j.cviu.2017.04.011

2) Validity and reliability of a computer-vision-based smartphone app for measuring barbell trajectory during the snatch by Carlos Balsalobre-Fernandez, Gretchen Geiser, Jon Krzyszkowski and Kristof Kipp.
https://doi-org.aurarialibrary.idm.oclc.org/10.1080/02640414.2020.1729453

3) A Method of Key Posture Detection and Motion Recognition in Sports Based on Deep Learning by Shaohong Pan.
https://doi.org/10.1155/2022/5168898

4) Recognizing Exercises and Counting Repetitions in Real Time by Talal Alatiah, Chen Chen. https://arxiv.org/pdf/2005.03194v1.pdf

9. Tutorials/Repositories that were Referenced

https://www.youtube.com/watch?v=5kaX3ta398w
https://www.youtube.com/watch?v=H7cGq0xIHbc&t=666s
https://www.youtube.com/watch?v=06TE_U21FK4