**COMP9444 Project Summary**

Gym Pose Detection and Classification

Neural Network Ninjas (NNN)

z5444643 Vishesh Malik

z5012904 Arifah binti Anwar Fadzil

z5465111 Sathwik Bharadwaj Udupi

z5447832 Devarshi Prakashbhai Patel

z5435266 Bhuvan Tulani

1. **Introduction**

Fitness and workout activities like squats, biceps curls and overhead press are some of the very common exercises people do in the gym. The "Real-Time Human Gym Pose Detection" is an innovative project that combines fitness tracking with artificial intelligence. The primary focus of this project is to enhance workout experience by utilizing multiple models including CNN Time Distributed (LRCN), LSTM and 3-D CNN to classify types of exercise and apply it to real time application while allowing the users to count repetitions of the exercise. For our validation part, we are using YOLO and MediaPipe to draw the bounding box on the video file before running the classification.

Firstly, the system that we developed will be able to recognize the type of exercise that the mp4 video file inputs. From there, we have generated a “h5” file of the model and used it in the real time application. The real time application can then count the repetitions of the exercise and come up with the accurate exercise prediction.

Counting exercise repetition is crucial during gym workout as it serves as metric to track the progress of a person in gym. Our system can be integrated into different areas including during gym workouts, fitness coaching sessions, and physiotherapy.

1. **Literature Review**

For this project, we extracted the key points which are already in the coco dataset from the json files. Keypoints are essential for accurately representing human body movements during exercises. A thorough comprehension of joint angles, body alignment, and movement trajectories is made possible by the usage of key points. In the context of our project, identifying the exercise the user is performing requires the integration of key points.

We also implement Yolo for object detection of videos. Yolo has demonstrated its potential to recognize human poses and track motions during exercises in fitness technology. In the paper “Human Pose Estimation and Action Recognition for Fitness Movements” published in 2023, YOLOv7 has managed to show its potential in action recognition especially when combined with other networks. [1] The efficiency of Yolo aligns with the requirements of our project, providing a robust solution for identifying and tracking individuals as they perform shoulder press, bicep curls, and squats.

In addition, we made use of Media Pipe, an open-source, cross-platform tool developed by Google that lets us run different machine learning models in real time. Its capacity to provide precise and reliable pose estimation across a range of body positions and orientations is its greatest asset. The framework creates a skeletal representation of the human body by offering a set of predetermined key landmarks. Understanding joint angles, body alignment, and movement trajectories during exercises requires knowledge of this specific information. After recognizing the exercise performed, the model will use preselected landmarks for the respective exercise. These landmarks contain the coordinates of the key points and then, the system calculates the angles of the major joint using vector dot product from the coordinates of the key points as shown in fig 1.

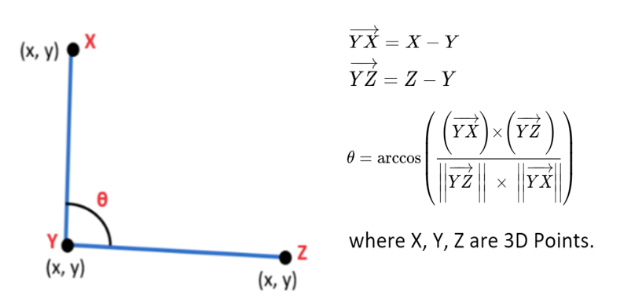


Fig 1: Calculating Angle Between Three Landmarks/key points. [2]

1. **Methods**

**3D CNN (Our best model)**

We used 3D Convolutional Neural Network (3D CNN) for the feature extraction and classification in this project. It is a robust and efficient solution for video data. Its effectiveness is attributed to the following key features. Firstly, two convo3D layers enable the model to capture spatial and temporal dependencies in the data. Using relu along with this has helped the model to learn nonlinear relationships (promoting faster convergence). Integrating it with max pooling 3D layers has helped in reducing the complexity and helps in focusing on salient features. After this, we have flattened the layers so that we can create fully connected layers which helps in extraction of high-level features. The following dense layers, employing softmax activation, facilitate multi class classification. All these implementations provided good interpretability and generalization capabilities to the model. We also tried varying the model by using tanh and sigmoid functions, but they were not giving satisfactory results. This might be since the gradients may become extremely small during backpropagation, which may lead to slow convergence, which may hamper the model to learn complex patterns.

**LRCN**

LRCN stands for long- term convolutional network and it is a combination of CNN and RNN. It was chosen as one of the methodologies for our project because of its ability to deal with large visual input like the videos we are using in our dataset. [3] In our LRCN model, we tried to improvise from [4] where we initialized the sequential model and stacked the time distributed layer with Conv2D. After tuning, the best model we decided to keep is the one with 16 filters and kernel size of 3 x 3 with relu activation function. The Time Distributed Wrapper allows the application of a layer to each frame of the input video. The Conv2D then detects the pattern in the input frames. This layer is then followed by Time Distributed Layer with Max Pooling 2D. This layer has a pooling window of 4 x 4 and it is used to reduce the spatial dimension of convolved features. A dropout layer is then added to prevent overfitting, and these layers are then followed by another 3 layers of Time Distributed Layer with Conv2D with 64 filters and each of these convolution layers are followed by maxpooling and the drop out layer. Before fitting this output into LSTM, we flattened it into the dimensional array. The final fully connected dense layer is then used to predict the classes. We then proceed to train the model with 50 epochs and early callback is added to prevent overfitting.

**Convo LSTM**

In the convo LSTM model, we implemented a sequential architecture which consists of ConvLSTM2D layers, maxpooling layers, dropout layers, and dense layers. We used Adam optimizer with a low learning rate. However, after training it only gave 40% -45% accuracy. The possible reasons of low accuracy may be due to the model might be overfitting and unable to detect complex patterns in the dataset. Using sigmoid may not have let the model converge properly. With the same input data preprocessing our other models, 3D CNN and LRCN have given higher accuracy.

1. **Experimental Setup**

We obtained the dataset from [InfiniteRep Dataset | Papers With Code.](https://paperswithcode.com/dataset/infiniterep) [5] There are 10 classes of exercises with each exercise giving 100 json and videos file. We decided to go with 3 type of exercises which are squat, biceps curls and overhead press and for each of the exercise dataset, it consists of 100 videos and corresponds json file where we then further split it into 80-10-10 for training- testing and validation. The json files consists of the bounding boxes coordinate, key points and skeletal using COCO format. After preprocessing using json file, each of the videos will then output 3 videos which we then input during training and testing. The videos are not taken from the same angle, and this proves to be one of the issues later for our prediction.

Some of the parameters that we can tune includes the number of strides, number of filters and filters size. The stride is the size of steps that the filter can move in each iteration. [6] The activation layer we have considered includes ReLU and sigmoid. Upon trial, we found out that the ReLu gives better accuracy for both our models. Although the accuracy we are getting is high, the classification on the validation layer is low. We then look into the Leaky ReLU, sigmoid and SoftMax function based on the explanation here in [7] but relu still gives the best result among all.

1. **Results**

The best 3D CNN models we had gave an accuracy of close to 100% and the ConvoLSTM gave the lowest accuracy.

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| 3DCNN | 100% |
| LRCN | 97% |
| Convo LSTM | 40% |

Table 1: Summary of accuracy for our models

From both the figures below, the trend is not smooth because of the angle of the camera in our input videos.

A graph of a curve

Description automatically generated with medium confidenceA graph with blue line and orange line

Description automatically generated

Fig 2: 3D CNN accuracy and loss plot

A graph of a graph with blue and orange lines

Description automatically generatedA graph with blue and orange lines

Description automatically generated

Fig 3: LRCN accuracy and loss plot

From the best model we have after tuning the parameters, we then summarize the classification performance.

|  |  |  |
| --- | --- | --- |
|  | **LRCN** | **3D CNN** |
|  | **Number of correct exercises predicted (Out of 10)** | |
| Squats | 7 | 9 |
| Biceps curls | 2 | 3 |
| Overhead press | 8 | 9 |

Table 2: Number of correct exercises predicted.

The exercise that has the highest error rate is biceps curls for both models. Overhead press and squats both gave excellent classification result. Comparing it to the result from the similar project literature we found in [8], their model accuracy is up to 96.87% when they are using SVM and a testing accuracy of 90.62%. One of the reasons why their model is performing better than ours is because of their dataset. The videos angle they are using is the same on all the same exercises hence making it easier for features extraction.

1. **Conclusion**

Our project has managed to achieve our goal which is to predict the correct exercise and incorporate it into a real time tracker. The best result we got was by using 3D CNN model where we managed to get an accuracy range between 97-99%. Our best model can also predict movement with an accuracy of 70%.

The strength of our system is shown when our best 3D CNN model is able to consistently predict squats and overhead press correctly with an accuracy of above 90% each time. For both our LRCN and 3D CNN models, the curls are the exercise that always give a lower accuracy score. In LRCN, the curls are always predicted as overhead press. This is expected as if we look into the similarity of the human posture when these two exercises are conducted. The only difference in differentiating between these two exercises lies only in the threshold angle we use. However, in the 3D CNN model, the curls are always wrongly predicted as squats. This is something that we fail to understand yet and would like to investigate more if time permits.

For the real time result, the real-time video classification we improvised from [9] is working fine but the video is lagging, and the classification is a bit slow. This may be due to two reasons: Firstly, our model is a bit heavy. Secondly, we don’t have ample computing power to manage the desired computing level required.

Lastly, for future work, we are contemplating working on some new features which can be used for a wider range of audiences. Some of the features we can include is adding more complex exercises for detection. Also, we are planning to integrate pose correction along with classification so that the user could see if he/she is going wrong and correct himself/herself accordingly. Additionally, the other two extra features we want to implement also include detecting multiple people and keeping a track of them individually and creating a customized rep counter for each exercise rather than having one counter for all exercises.

1. **References**

**[1]** H. Fu, J. Gao, and H. Liu, “Human pose estimation and action recognition for fitness movements,” *Computers & Graphics*, vol. 116, pp. 418–426, 2023, DOI: <https://doi.org/10.1016/j.cag.2023.09.008>

**[2]** M. Murugavel. (2019). *Find the Angle between three points from 2D using python* [Online]. Available: <https://manivannan-ai.medium.com/find-the-angle-between-three-points-from-2d-using-python-348c513e2cd>.

**[3]** Taha Anwar, Rizwan Naeem and Momin Anjum (2021). *Human Activity Recognition using TensorFlow CNN-LSTM.* Bleed AI Academy. Available: <https://bleedaiacademy.com/human-activity-recognition-using-tensorflow-cnn-lstm/>.

### **[4]** ByungSoo Ko *(2017). Long-term Recurrent Convolutional Network (LRCN)*. Kobiso GitHub Pages. Available: <https://kobiso.github.io/research/research-lrcn/>.

**[5] *I****nfiniteRep Dataset*. Papers with Code. Available: <https://paperswithcode.com/dataset/infiniterep>.

**[6]** Mukesh, C. (2020). *Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax*. Medium. Available: <https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>.

**[7]** Chengping Rao & Yang Liu. (2020). *Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization.* Dept. of Mechanical and Industrial Eng., Northeastern Uni, Boston, MA 02115, USA. Available: <https://doi.org/10.1016/j.commatsci.2020.109850>.

**[8]** A. Rahmadani, B. S. Bayu Dewantara and D. M. Sari, "Human Pose Estimation for Fitness Exercise Movement Correction," 2022 International Electronics Symposium (IES), Surabaya, Indonesia, 2022, pp. 484-490, DOI: [10.1109/IES55876.2022.9888451](https://doi.org/10.1109/IES55876.2022.9888451)

**[9]** Prasanna, C. (2022). *Exercise\_Recognition\_AI.* GitHub.

[Online]. Available: [GitHub - chrisprasanna/Exercise\_Recognition\_AI: Developing a virtual personal fitness tracker and exercise activity recognition A.I. using computer vision and deep learning.](https://github.com/chrisprasanna/Exercise_Recognition_AI)