AI Coach App for Gym

Faculty of Computers and Information

Assiut University

Table of contents

1. Team Information
2. Project Description
3. Model 1: LRCN Model
   - Demo
   - Contribution
   - Data
   - Project Architecture
   - Methods
   - Results

Table of contents (cont.)

4 Model 2: ConvLSTM Model with Pose Estimation
- Demo
- Contribution
- Project Architecture
- Methods
- Results

Team Information

Team ID:	1
ID	Team Names
162021323	مروه عامر مرسي
162021331	مريم عادل شحاته
162021330	مريم حسن عبد الشفيق

Table of contents

1. Team Information
2. Project Description
3. Model 1: LRCN Model
   - Demo
   - Contribution
   - Data
   - Project Architecture
   - Methods
   - Results

Table of contents (cont.)

4 Model 2: ConvLSTM Model with Pose Estimation
- Demo
- Contribution
- Project Architecture
- Methods
- Results

Project Description

AI Coach Application for Gym that takes a video stream of exercises and classify whether they are correct or not using Deep Learning
This model recognizes 2 exercises which are:

- Front Raise
- Shoulder Press

We use Two models:

- LRCN Model
- CONVLSTM Model with Pose Estimation

Table of contents

1. Team Information
2. Project Description
3. Model 1: LRCN Model
   - Demo
   - Contribution
   - Data
   - Project Architecture
   - Methods
   - Results

Table of contents (cont.)

4 Model 2: ConvLSTM Model with Pose Estimation
- Demo
- Contribution
- Project Architecture
- Methods
- Results

Demo

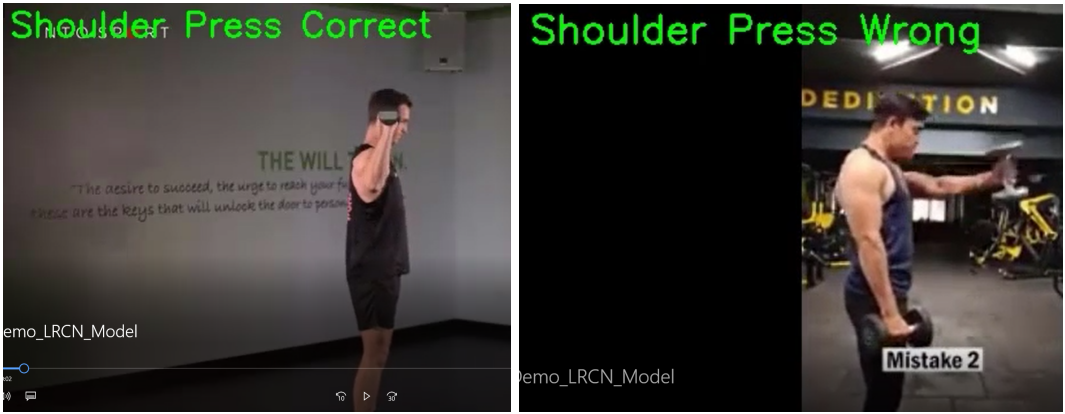


Fig.1. Demo

https://github.com/Mariam-Hsn/AI_Coach_app_for_Gym

Contribution

- Found an exercises video dataset from Kaggle
- Got a model from YouTube video
- Edited the training to test ratio from [75%,25%] to [80%,20%]
- Edited the model to recognize 2 exercises (Correct and Wrong) instead of 4 actions
- Added some layers to achieve best Accuracy
- Added the confusion Matrix
- Added a python app for testing

Data

We take two exercises (Front Raise and Shoulder Press) from a video dataset from kaggle that was uploaded by **Jiunn**
All videos are RGB
Each folder in the dataset represents a Correct or Wrong Exercise
The Correct Front Raise folder consists of 50 videos
The Wrong Front Raise folder consists of 229 videos
The Correct Shoulder Press folder consists of 59 videos.
The Wrong Shoulder Press folder consists of 217 videos.
The Same Dataset will be used for the 2 models

Data (cont.)

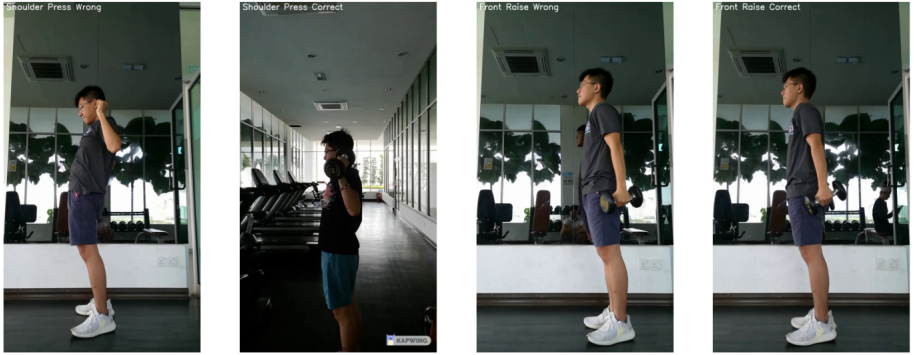


Fig.2. Data Visualization

dataset from this link: `https://www.kaggle.com/datasets/jiunn1998/workout-exercise?resource=download`

Project Architecture

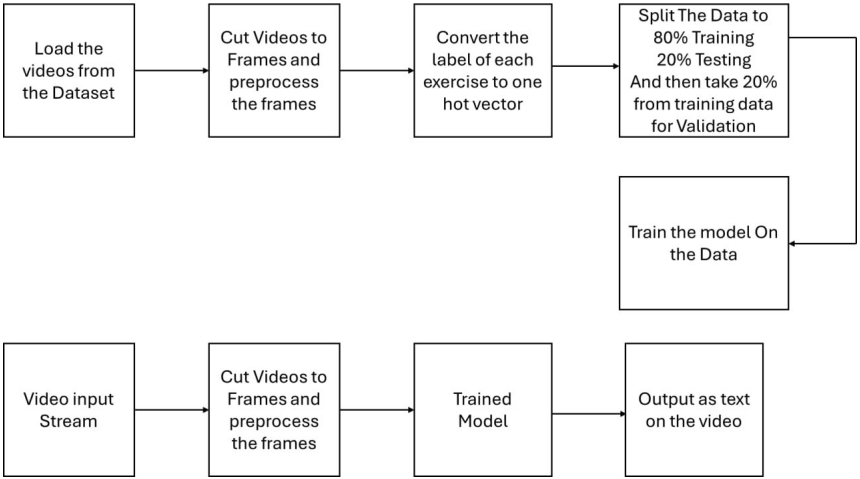


Fig.3. Model 1 Architecture

Methods

The model consists of the following layers:

Layer (type)	Output Shape	Param #
time_distributed_223 (TimeDistributed)	(None, 20, 64, 64, 16)	448
time_distributed_224 (TimeDistributed)	(None, 20, 64, 64, 32)	4640
time_distributed_225 (TimeDistributed)	(None, 20, 16, 16, 32)	0
time_distributed_226 (TimeDistributed)	(None, 20, 16, 16, 32)	0
time_distributed_227 (TimeDistributed)	(None, 20, 16, 16, 64)	18496
time_distributed_228 (TimeDistributed)	(None, 20, 4, 4, 64)	0
time_distributed_229 (TimeDistributed)	(None, 20, 4, 4, 64)	0
time_distributed_230 (TimeDistributed)	(None, 20, 4, 4, 64)	36928
time_distributed_231 (TimeDistributed)	(None, 20, 2, 2, 64)	0
time_distributed_232 (TimeDistributed)	(None, 20, 2, 2, 64)	0
time_distributed_233 (TimeDistributed)	(None, 20, 2, 2, 128)	73856
time_distributed_234 (TimeDistributed)	(None, 20, 1, 1, 128)	0
time_distributed_235 (TimeDistributed)	(None, 20, 128)	0
lstm_7 (LSTM)	(None, 32)	20608
dense_14 (Dense)	(None, 4)	132

Total params: 155108 (605.89 KB)
Trainable params: 155108 (605.89 KB)
Non-trainable params: 0 (0.00 Byte)

Fig.4. Model Layers

Methods (cont.)

The Optimizer used is : "adam"
The Loss Function used is : "categorical crossentropy"

Results

We Show the results using 3 measures:**Accuracy, Loss and Confusion Matrix**

```
# Evaluate the trained model.
model_evaluation_history = LRCN_model.evaluate(features_test, labels_test)
```

```
4/4 [==============================] - 2s 128ms/step - loss: 0.0054 - accuracy: 1.0000
```

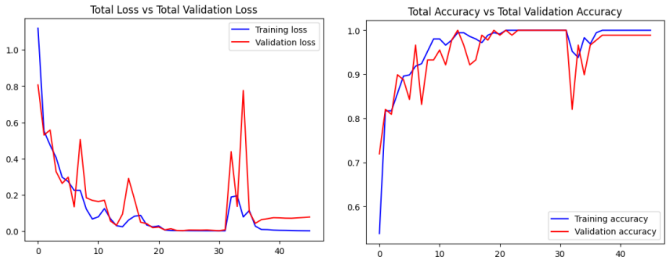


Fig.5. Results

Results (cont.)

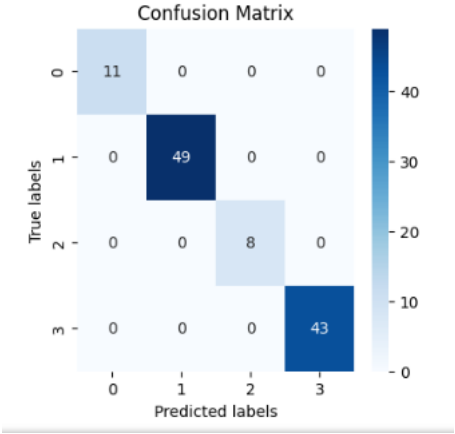


Fig.6. Confusion Matrix

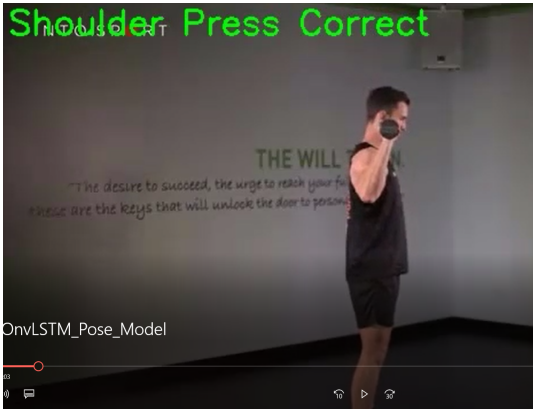
Table of contents

1. Team Information
2. Project Description
3. Model 1: LRCN Model
   - Demo
   - Contribution
   - Data
   - Project Architecture
   - Methods
   - Results

Table of contents (cont.)

4 Model 2: ConvLSTM Model with Pose Estimation
- Demo
- Contribution
- Project Architecture
- Methods
- Results

Demo



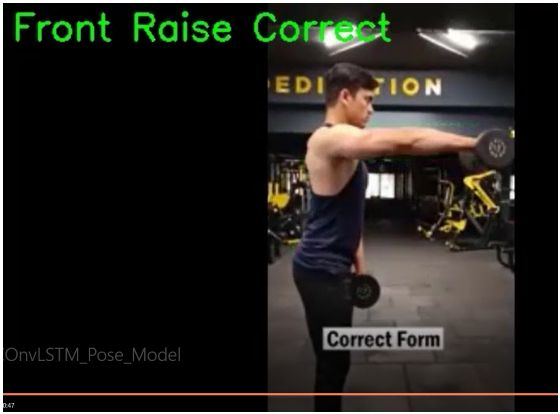


Fig.7. Demo

https://github.com/Mariam-Hsn/AI_Coach_app_for_Gym

Contribution

- Edited The ConvLSTM Model to achieve best Accuracy
- extracted landmarks using mediapipe pose estimation
- Trained the model on the dataset with landmarks
- changed the number of epochs from 50 to 20
- Edited the model to recognize 2 exercises (Correct and Wrong) instead of 4 actions
- Added the Confusion matrix

Project Architecture

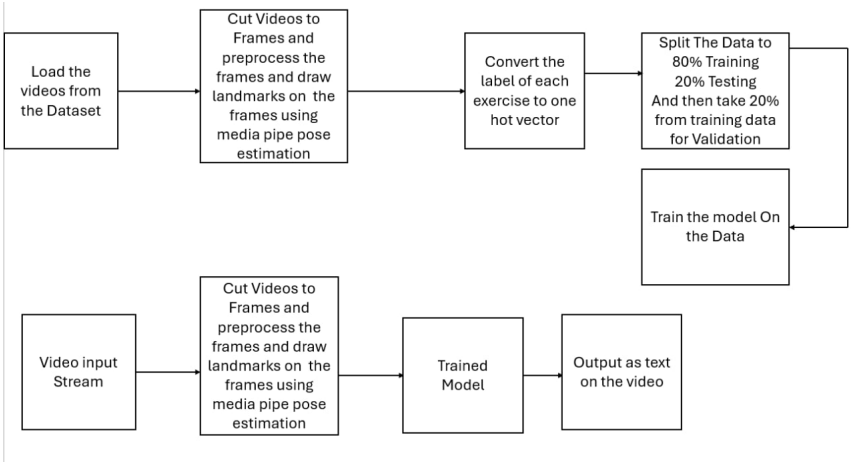


Fig.8.Project Architecture

Methods

The model consists of the following layers:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv_lstm2d (ConvLSTM2D)    (None, 20, 148, 148, 4)   1024

 max_pooling3d (MaxPooling3  (None, 20, 74, 74, 4)     0
 D)

 time_distributed (TimeDist  (None, 20, 74, 74, 4)     0
 ributed)

 conv_lstm2d_1 (ConvLSTM2D)  (None, 20, 72, 72, 8)     3488

 max_pooling3d_1 (MaxPoolin  (None, 20, 36, 36, 8)     0
 g3D)

 time_distributed_1 (TimeDi  (None, 20, 36, 36, 8)     0
 stributed)

 conv_lstm2d_2 (ConvLSTM2D)  (None, 20, 34, 34, 14)    11144

 max_pooling3d_2 (MaxPoolin  (None, 20, 17, 17, 14)    0
 g3D)

 time_distributed_2 (TimeDi  (None, 20, 17, 17, 14)    0
 stributed)

 conv_lstm2d_3 (ConvLSTM2D)  (None, 20, 15, 15, 16)    17344

 max_pooling3d_3 (MaxPoolin  (None, 20, 8, 8, 16)      0
 g3D)

 flatten (Flatten)           (None, 20480)             0

 dense (Dense)               (None, 4)                 81924

=================================================================
Total params: 114924 (448.92 KB)
Trainable params: 114924 (448.92 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Model Created Successfully!
```

Fig.9. Model Layers

Methods

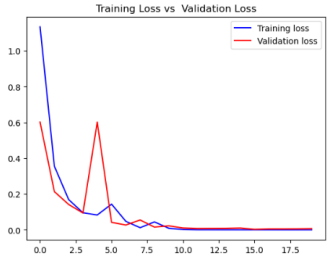
The Optimizer used is : "adam"
The Loss Function used is : "categorical crossentropy"

Results

We Show the results using 3 measures:**Accuracy, Loss and Confusion Matrix**

```
# Evaluate the trained model.
model_evaluation_history = pose_model.evaluate(features_test, labels_test)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 4s 972ms/step - accuracy: 1.0000 - loss: 0.0028
```



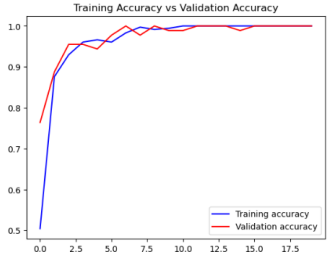


Fig.10. Results

Results (cont.)

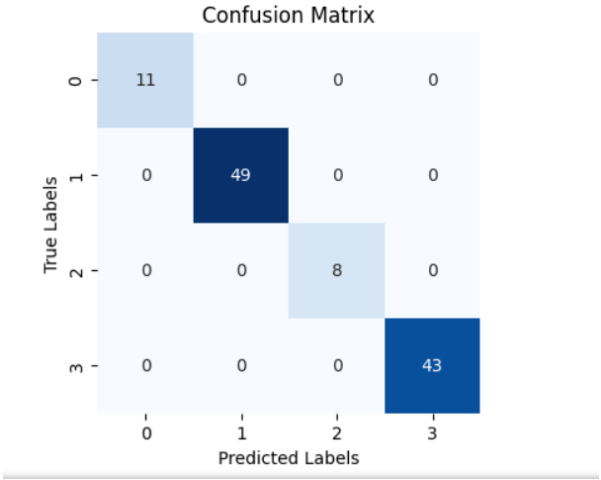


Fig.6. Confusion Matrix