*electronics*

MDPI

*Article*

# Automatic Evaluation Method for Functional Movement Screening Based on Multi-Scale Lightweight 3D Convolution and an Encoder–Decoder

**Xiuchun Lin [1], Yichao Liu [2], Chen Feng [3], Zhide Chen [2,*], Xu Yang [4,*] and Hui Cui [5]**

[1] Fujian Institute of Education, Fuzhou 350025, China; qsz20231889@student.fjnu.edu.cn
[2] College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China; qsx20221335@student.fjnu.edu.cn
[3] Department of Information Engineering, Fuzhou Polytechnic, Fuzhou 350003, China; fengchen@fvti.edu.cn
[4] College of Computer and Data Science, Minjiang University, Fuzhou 350108, China
[5] Department of Software Systems & Cybersecurity, Monash University, Melbourne, VIC 3800, Australia; hui.cui@monash.edu
* Correspondence: zhidechen@fjnu.edu.cn (Z.C.); xu.yang@mju.edu.cn (X.Y.)

**Abstract:** Functional Movement Screening (FMS) is a test used to evaluate fundamental movement patterns in the human body and identify functional limitations. However, the challenge of carrying out an automated assessment of FMS is that complex human movements are difficult to model accurately and efficiently. To address this challenge, this paper proposes an automatic evaluation method for FMS based on a multi-scale lightweight 3D convolution encoder–decoder (ML3D-ED) architecture. This method adopts a self-built multi-scale lightweight 3D convolution architecture to extract features from videos. The extracted features are then processed using an encoder–decoder architecture and probabilistic integration technique to effectively predict the final score distribution. This architecture, compared with the traditional Two-Stream Inflated 3D ConvNet (I3D) network, offers a better performance and accuracy in capturing advanced human movement features in temporal and spatial dimensions. Specifically, the ML3D-ED backbone network reduces the number of parameters by 59.5% and the computational cost by 77.7% when compared to I3D. Experiments have shown that ML3D-ED achieves an accuracy of 93.33% on public datasets, demonstrating an improvement of approximately 9% over the best existing method. This outcome demonstrates the effectiveness of and advancements made by the ML3D-ED architecture and probabilistic integration technique in extracting advanced human movement features and evaluating functional movements.

**Keywords:** functional movement screening; human movement feature; 3D convolution; encoder–decoder; automatic evaluation method



**Citation:** Lin, X.; Liu, Y.; Feng, C.; Chen, Z.; Yang, X.; Cui, H. Automatic Evaluation Method for Functional Movement Screening Based on Multi-Scale Lightweight 3D Convolution and an Encoder–Decoder. *Electronics* **2024**, *13*, 1813. https://doi.org/10.3390/electronics13101813

Academic Editor: Francesco Beritelli

Received: 9 April 2024
Revised: 27 April 2024
Accepted: 2 May 2024
Published: 7 May 2024



## 1. Introduction

Functional Movement Screening (FMS) assesses an individual's movement abilities and identifies potential risks for the occurrence of sports injuries. It analyzes the body's flexibility and stability by assessing basic movement patterns. FMS includes seven test actions and three exclusion actions. The seven test actions are Deep Squat, Active Straight Leg Raise, Trunk Stability-Push Up, Hurdle Step, Shoulder Mobility, In-Line Lunge, and Rotary Stability-Quadruped. The three exclusionary actions are the Prone Press-up Test, Impingement Test, and Kneelinglumbar Test. Although many individuals demonstrate excellent athletic ability, some cannot perform specific movements effectively on the FMS assessment, resulting in lower scores. These people tend to use compensatory movements to complete specific movements. If this compensation continues for a long time, it may cause the body to become accustomed to non-standard movement patterns, thereby affecting the balanced development of the body and increasing the risk of injury. The FMS is used to score each of the subject's movements to discover the areas with the most severe movement

defects, which can improve functional movements, improve sports performance, and reduce the risk of sports injuries in a targeted manner. The main evaluation method usually used in FMS is expert on-site evaluation, which involves experts observing each subject's sports performance. However, this method has some drawbacks, including being time-consuming and labor-intensive and susceptible to experts' subjective opinions, which may reduce the accuracy of the assessment's results.

With the development of deep learning, Convolutional Neural Networks (CNNs) have been widely applied to understanding actions and have made significant advancements compared to traditional methods [1]. Each step of a CNN involves three fundamental operations: convolution, the nonlinear activation of neurons, and feature pooling [2]. In the work by Baccouche et al. [3], a 3D CNN treats the input as a spatiotemporal volume. Subsequently, the features extracted by the 3D CNN are trained within a Long Short-Term Memory (LSTM) network. Zhou et al. [4] coupled 3D input feature maps with 2D convolutional blocks in a block-serial manner. They also added connections that incorporate cross-domain residual methods along the temporal dimension to better extract temporal information and reduce complexity.

The rapid development of deep learning for action recognition and quality assessment has promoted the progress of FMS. Leveraging these advancements, Andreas et al. [5] proposed a Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) model for classifying functional movements. Similarly, Duan et al. [6] utilized a CNN model to classify electromyography (EMG) signals associated with functional movements, achieving impressive accuracies for movements such as squat, step, and straight squat. Deep learning algorithms excel in automatically extracting movement features, thereby enhancing the accuracy of movement recognition. However, their efficacy is contingent upon the availability of a vast amount of training data, which can be prohibitively time-consuming to acquire. Additionally, many traditional deep learning methods rely on feature extraction by models like I3D [7–10]. I3D employs three-dimensional convolutional kernels, which implies that each kernel provides both spatial and temporal information. Consequently, the number of parameters for each convolutional kernel is a multiple of those in a two-dimensional kernel, leading to a substantial increase in the parameter count of the I3D model.

Alternatively, machine learning approaches have demonstrated promise, particularly in movement quality evaluation. For instance, an automatic AdaBoost-based FMS evaluation method introduced by Wu et al. [11] utilizes multiple weak classifiers to construct a robust classifier. Similarly, Bochniewicz et al. [12] employed a random forest model to evaluate arm movements in stroke patients, employing a minority–majority voting mechanism for classification label prediction. These methods require less data and offer interpretability through manual feature extraction. Moreover, Hong et al. [13] demonstrated the feasibility of using depth cameras in FMS assessment by collecting a dataset using Azure Kinect depth sensors. They improved the accuracy of FMS assessment by forming a robust classifier that combines three Gaussian mixture models, each trained on datasets with different scores.

Given the above shortcomings, this paper proposes an automatic evaluation method for FMS based on multi-scale lightweight 3D convolution and an encoder–decoder(ML3D-ED). This method employs ML3D to extract features from videos. It combines the ED structure to learn the score distribution features of individual movements, thus improving the accuracy and reliability of the evaluation. The main contributions of this paper are as follows:

1. In this paper, an ML3D module is designed as an alternative to the I3D feature extraction module. Compared with the I3D model, the parameters and computation (floating point operations per second, FLOPs) of the ML3D-ED model were reduced by 59.55% and 77.67%, respectively.
2. This paper proposes an ED structure network to process features extracted by the ML3D module, learn subtle movement changes in advanced movement quality evaluation, apply it to functional movement screening, and improve the accuracy of the evaluation results.

3. The paper employs a score prediction approach to transform label data processed by the ED into a distribution of scores. Utilizing a Gaussian distribution, it compares losses between true and predicted values for samples. Compared to the current most popular approach, the accuracy of this this method has been improved by nearly 9%.

## 2. Relevant Theories

### *2.1. Functional Movement Screening*

FMS is a tool used to assess an individual's movement abilities and potential risks in their movements. It involves observing whether a subject is stable in executing these movements and whether there are any abnormalities in their execution of movements. The core concept of FMS is to reveal potential problems and imbalances in an individual's movement by evaluating a series of basic, functionally significant movements. Through FMS, we are able to identify potential factors that may contribute to sports injuries, such as muscle imbalance, joint instability, or movement skill deficits. This systematic evaluation helps an individual develop a personalized training plan that emphasizes the improvement of individual weaknesses, and provides specific recommendations for exercises that can be used to maximize functional gains.

The FMS process includes a series of rigorous movement tests, such as deep squat, walk, and rotate. It involves observing a subject's performance in executing these movements to evaluate their flexibility, stability, and coordination. This comprehensive evaluation allows sports professionals to deeply understand an individual's physical function and provides important information for developing a targeted rehabilitation plan or training program.

### *2.2. Video-Based Action Quality Evaluation*

The purpose of video-based action quality assessment (AQA) is to detect and assess the completion of actions in a video. In quality score-based evaluation methods, videos to be evaluated are usually segmented into appropriate clip-level or frame-level data. Next, these data are processed through the feature extraction module to extract feature vectors related to action features. The feature vectors will serve as inputs for regression or classification functions, yielding quality assessment scores accordingly.

Bai et al. [14] proposed a temporal decoder method for video-based action quality assessment, but the lack of labels may affect its performance. Gordon [15] explored a body center-of-mass trajectory-based scoring method in small-scale applications, but it needs to be validated in wider applications. The key segment extraction system proposed by Li et al. [16] only extracts part of the scores, which is inconsistent with the diving rules. The hidden Markov model-based hierarchical classification method proposed by Tao et al. [17] uses small datasets and has a limited generalization ability.

Parmar et al. [18] used methods such as the support vector machine, neural network, and enhanced decision tree to classify physiotherapeutic rehabilitation data, but the data samples were too abundant. The multi-scale convolutional LSTM network proposed by Xu et al. [19] is suitable for figure skating, but its complex background may lead to significant prediction errors.

### *2.3. I3D Architecture*

The continuous improvement and application of I3D-LSTM architecture in the field of video analysis have shown its excellent performance in capturing important quality-related information. Carreira et al. [20] proposed this architecture in their original research and continued to improve it in subsequent studies. Through empirical validation using the Kinetics dataset, researchers successfully demonstrated the high performance of I3D-LSTM in a wide range of action classes [21,22]. Hara et al. [23] explored the development of 3D CNNs in the field of videos in depth, focusing on its relationship with 2D CNNs and ImageNet. These studies have provided a broader context for understanding the motivations and development behind I3D-LSTM. Wang et al. [24] proposed the I3D-LSTM

network architecture, which cleverly combines the I3D architecture and the long short-term memory (LSTM) network. This integration was designed to improve the recognition of human movements in videos. Through a comprehensive utilization of spatiotemporal information, the I3D-LSTM architecture significantly improves the accuracy of action recognition tasks, bringing new prospects to the fields of video analysis and human–computer interaction.

We have replaced the widely used I3D module with an indigenously designed lightweight module. This improvement allows the system to significantly reduce the model's complexity and the computation costs while maintaining its performance. Our new module combines advanced deep learning techniques with a carefully designed architecture to offer greater efficiency and flexibility to the system.

## 3. The Protocol Proposed in This Paper

The comprehensive network model proposed in this study, consisting of a multi-scale lightweight 3D convolutional network (ML3D) and an encoder–decoder (ED), is used to evaluate FMS. As shown in Figure 1, the ML3D network takes a sequence of video frames as an input and considers video features in both temporal and spatial dimensions through lightweight 3D convolutional operations. By extracting features at different scales and levels, the ML3D network is able to capture the spatiotemporal dependencies and feature expression capabilities of videos. Subsequently, the features extracted by ML3D are passed to the ED. The input data are non-linearly represented via multiple 1D convolutions. Finally, the predicted distribution is obtained by performing a probabilistic integration on the features of different scales output by the ED. This predicted distribution represents the analysis and learning of input videos by the model for prediction of the confidence of each class.

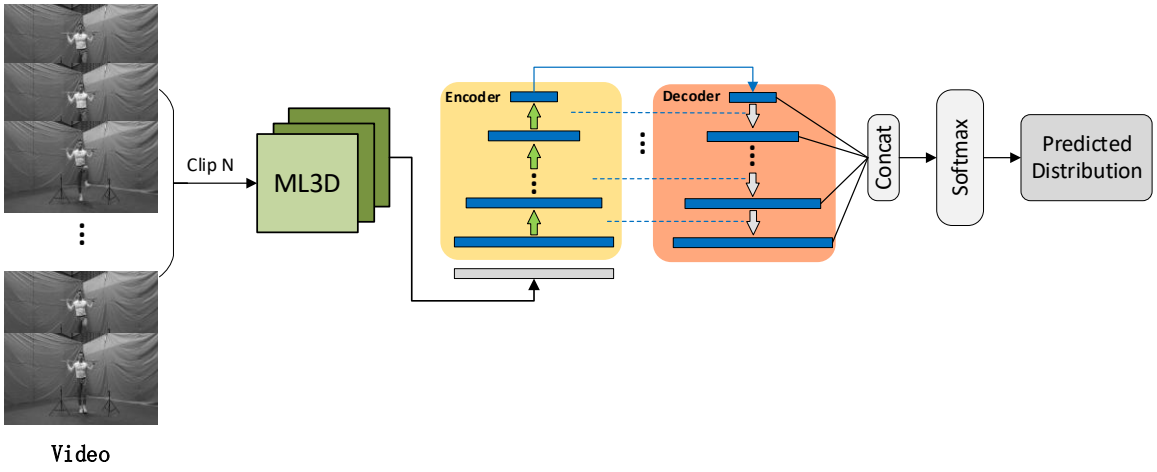


**Figure 1.** ML3D-ED network architecture.

### *3.1. Data Preprocessing*

There are few video frames in the FMS data, which means that the criteria required by the model for the number of frames cannot be met, leading to a failure to obtain effective features. To cope with this situation, a linear interpolation-based video frame interpolation method is used in this protocol to interpolate between adjacent frames to generate additional interpolated frames.

First, two adjacent frames are selected as reference frames before and after the missing frame in the video sequence. These two frames contain most of the visual information required for the missing frame. This process involves using linear interpolation techniques to take the weighted sum of two adjacent image frames and then generate a series of evenly spaced interpolation frames between the two adjacent frames. Ultimately, the process

involves calculating the linear relationship between pixels to derive the pixel value of the interpolated frames.

$$I = (1 - a) \times v_1 + a \times v_2 \tag{1}$$

The parameter $a$ serves as the interpolation factor, enabling adjustments to control the weighted ratio between two given values. Denoted by $v_1$ and $v_2$, these values correspond to the interpolation target and the values of neighboring images. These values encompass pixel data, color components, or other feature information necessitating frame interpolation, particularly in video interpolation. At the pixel level, they denote RGB color channel values per pixel. At the feature level, they may indicate feature points or vectors. A pixel-level interpolation technique synthesizes new frames in the preprocessing phase by averaging adjacent pixels. This process ensures seamless transitions between frames and augments the frame count.

### *3.2. ML3D*

A natural method to encode spatiotemporal information in videos involves extending the convolution kernels in a CNN from 2D to 3D, enabling the training of a new 3D CNN. This approach allows the network to learn both the visual appearance within individual video frames and the temporal evolution across frames. However, despite demonstrating superior performance in recent studies, training 3D CNNs is computationally demanding and requires significant computational resources. Taking the widely used 11-layer 3D CNNs (namely, the C3D [25] network) as an example, the model size reaches 321 MB, even larger than the 152-layer 2D ResNet (ResNet-152) [26] (235 MB), which makes training a very deep 3D CNN extremely difficult.

Three-dimensional convolution is equivalent to simultaneously convolving two-dimensional feature maps from multiple time steps. Therefore, assuming that the size of the 3D convolutional filter is $d \times k \times k$, it can be decoupled into a $1 \times k \times k$ convolutional filter equivalent to the 2D convolutional filter in the spatial domain and a $d \times 1 \times 1$ convolutional filter customized in the temporal domain, as shown in Figure 2.

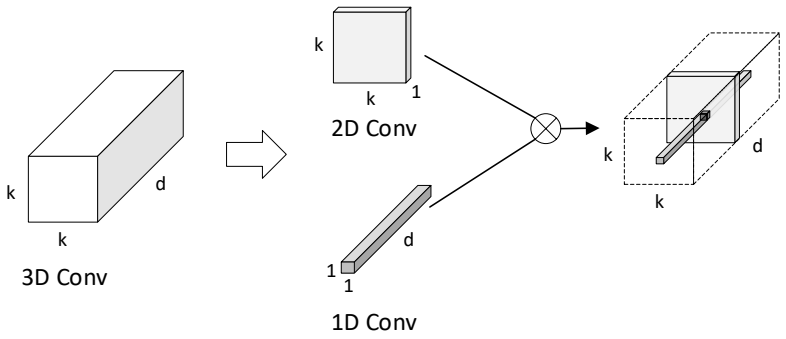


**Figure 2.** Three-dimensional filter equivalently transformed into two-dimensional + one-dimensional filters.

This decoupling can significantly reduce the number of model parameters, and subsequent experiments have proven that this decoupling can effectively extract feature information from videos. In this paper, multiple 2D convolution kernels of different sizes are used to extract information at different scales, and then $1 \times 3 \times 3$ convolutional filters and $3 \times 1 \times 1$ convolutional filters are used in the spatial domain and temporal dimension, respectively, for equivalent lightweight 3D convolution (L3D). The overall framework adopts a residual learning approach, in which the raw input is downsampled through the downsample module and aligned with the feature dimensions output by the L3D convolution, as shown in Figure 3.

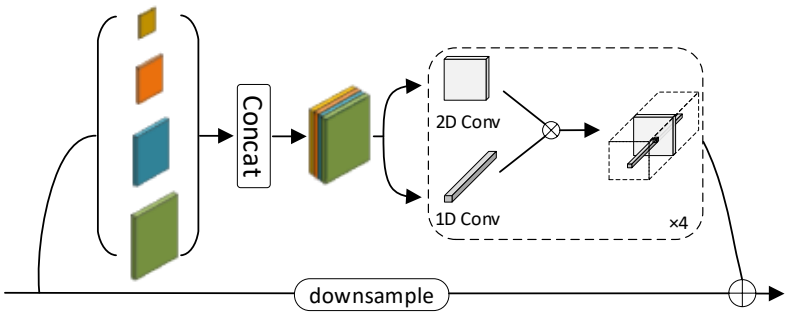


**Figure 3.** ML3D architecture.

### *3.3. Encoder–Decoder*

In this paper, an ED architecture is built to learn the features extracted by ML3D to obtain the final score prediction, as shown in Figure 4. Its basic structure consists of two parts: an encoder and a decoder. The encoder adopts a stepwise downsampling convolution architecture to map the input sequence into a latent space, and the decoder generates a target sequence in this space. The ED model can encode variable-length sequences into fixed-length state vectors in the encoding stage and then decode the state vectors and generate variable-length prediction sequences in the decoding stage. The decoder and encoder skip connections to achieve multi-scale feature fusion, effectively solving the error accumulation problem.

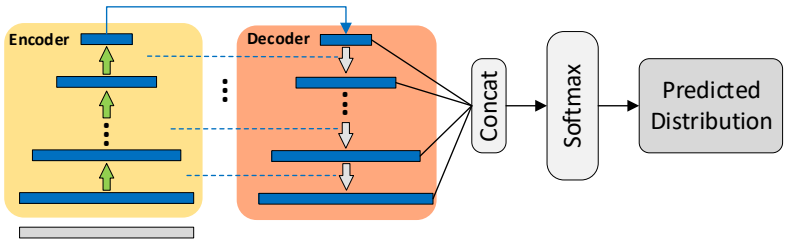


**Figure 4.** Encoder–decoder.

The outputs of multiple decoders with different resolutions are concatenated and then the final score distribution prediction is output through the Softmax function.

### *3.4. Score Prediction*

Given the difference between video-based action quality assessments and image recognition, there are similar image features between every two adjacent frames. Therefore, processing features learned by ML3D and ED models enhance the accuracy and reliability of ratings. We adopted a method that incorporates considering uncertainty in the scoring process. This approach enables a more comprehensive capturing of the variations and fluctuations in movement quality, ensuring that the ratings accurately reflect the actual quality of the movements and account for any instability.

#### 3.4.1. Gaussian Distribution of the Initial Data

In the final layer of the algorithm architecture shown in Figure 1, there are four output nodes corresponding to the four different levels of FMS scores. The video features are transformed into a score distribution during the data processing stage. To process these score distributions, we apply a Gaussian function to smooth them. Equation (2) describes the probability density function values of the real scores, which are used to convert the discrete score data into a continuous probability distribution. This enables the model to learn and predict more refined rating outcomes.

$$g(c) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(c-s)^2}{2\sigma^2}) \quad (2)$$

$s$ represents the mathematical expectation of the label score, and $\sigma$ represents the standard deviation. The scores are discretized at intervals as $c = [c_1, c_2, \ldots, c_m]$. The magnitude of each score is described as $g(c) = [g(c_1), g(c_2), \ldots, g(c_m)]$, representing probability density values. Next, we normalize the function to obtain normalized probability values. This step is taken to facilitate the optimization of image loss from the neural network in subsequent calculations. We normalize the probability density function values to obtain normalized probability values.

$$tmp_i = \frac{g(c_i)}{\sum_{j=1}^{m} g(c_j)} \quad (3)$$

$m$ is the number of classes and is derived from the normalized probability value, which helps subsequent neural networks to learn to calculate image loss for optimization.

#### 3.4.2. Kullback–Leibler (KL) Divergence

After inputting the image to the ML3D-ED model and processing it, the model outputs $m$ categories. The softmax layer converts the output value into a probability distribution score, $pred = [pred_1, pred_2, \ldots, pred_m]$. KL divergence measures the matching degree between two distributions, $tmp$ and $pred$. The calculation method for KL divergence is shown in Formula (4).

$$KL\{\text{tmp} \parallel \text{pred}\} = \sum_{i=1}^{n} tmp_i \log \frac{tmp_i}{pred_i} \quad (4)$$

Finally, when making predictions, the final predicted class is determined by selecting the value with the highest probability among the prediction scores.

## 4. Experiment

### *4.1. Data and Experimental Environment*

This paper uses the dataset created by Xing et al. [27], comprising various movements, including the deep squat, hurdle, split squat, shoulder mobility, active straight leg, raise rotary stability-quadruped, and trunk stability push-up. Each movement is performed on both the left and right sides. The dataset was collected from 45 individuals across different age groups (18 to 59 years old), with annotations provided by three FMS experts. Scores assigned to each movement range from 0 to 3.

The experimental environment is as follows: Intel(R) Xeon(R) Silver 4310 CPU @ 2.10 GHz processor, 26 G memory. GPU: RTX 4090 (24 GB) × 4, PyTorch v1.13.1, Python v3.9 (ubuntu22.04). Table 1 describes the dataset composition structure of each movement, including the training set and test set of the movement, as well as the frequencies of each movement for 1, 2, and 3 points.

**Table 1.** Number of single movements with different scores.

| | Training Set | | | Test Set | | |
|---|---|---|---|---|---|---|
| **ID** | **1** | **2** | **3** | **1** | **2** | **3** |
| M01 | 13 | 69 | 17 | 4 | 23 | 5 |
| M03 | 28 | 54 | 18 | 9 | 18 | 8 |
| M05 | 8 | 75 | 17 | 2 | 25 | 8 |
| M07 | 18 | 9 | 5 | 6 | 3 | 2 |
| M09 | 9 | 54 | 39 | 3 | 18 | 12 |
| M11 | 7 | 88 | 9 | 3 | 18 | 12 |
| M12 | 3 | 77 | 8 | 2 | 26 | 3 |
| M14 | 6 | 88 | 1 | 2 | 28 | 1 |

### *4.2. Evaluation Metrics*

The evaluation metrics in this paper include accuracy, macro *F*1, and Kappa coefficient.

1. Accuracy: This represents the effectiveness of the model's predictions. It is the ratio of the sum of the number of samples predicted to be correct to the total number of samples, as shown in Formula (5).

$$p_0 = \frac{\sum_{i=1}^{C} T_i}{n} \tag{5}$$

where $C$ is the number of classes, $T_{\mathrm{i}}$ is the number of samples classified correctly in $i$ th class, and $n$ is the number of overall samples.

2. Macroscopic F1 (macro _F1): This is used to measure the accuracy of multiclass classification. The prerequisite for calculating macro_F1 is to calculate F1_Score, which can be derived from Formula (6). It is a measure of classification tasks and is defined as the harmonic mean of precision and recall. Then, macro_F1 is calculated based on the value of F1_Score and Formula (7).

$$Fl\_score_{\mathrm{i}} = 2\frac{Recall_i \times Precision_i}{Recall_i + Precision_i} \tag{6}$$

In the above formula, $Recall_i$ is the recall of the $i$ th class, and $Precison_i$ is the precision of the $i$ th class.

$$\text{macro\_F1} = \frac{\sum_{i=l}^{C} Fl\_score_i}{C} \tag{7}$$

In the above formula, $C$ is the number of classes.

3. Kappa coefficient: This is used to measure agreement and can also be used as a measure of precision. For classification tasks, agreement is defined as the degree of consistency between the model prediction results and the actual classification results. The calculation of the Kappa coefficient is based on the confusion matrix. It has a value between $-1$ and 1, and is usually greater than 0, which is shown in Formula (8):

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e} \tag{8}$$

In Formula (8), $p_o$ is the accuracy and is consistent with Formula (8). $p_e$ represents the accidental agreement, derived from Formula (9):

$$p_e = \frac{\sum_{i=1}^{c} a_i \times b_i}{n} \tag{9}$$

In the above formula, $a_i$ is the number of actual samples of the th class, and $b_i$ is the number of predicted samples of the $i$ th class. $c_i$ is the total number of classes, and $n$ is the total number of samples.

### *4.3. Experiment and Result Analysis*

The experimental hyperparameters are set as follows: the batch size is 8, the epoch is 150, the initial learning rate is $10^{-4}$, and the gradient optimization algorithm is adam. The dataset is divided into training and testing sets, and the ratio of the training set to the test set is 3:1.

#### 4.3.1. Comparative Experiment Analysis

The superiority of this method is validated by comparing it with advanced video-based quality assessment algorithms using five evaluation metrics: accuracy, macro F1 score, Kappa coefficient, model parameters, and computational complexity. The experimental comparison data are shown in Table 2.

**Table 2.** Comparison of the experimental data.

| Model | Accuracy/% | maF1/% | Kappa/% |
|---|---|---|---|
| Improved GMM [13] | 80.00 | 77.00 | 67.00 |
| C3D-LSTM [28] | 74.44 | 74.35 | 61.66 |
| I3D-LSTM | 71.11 | 70.90 | 56.66 |
| I3D-MLP [29] | 84.44 | 84.53 | 76.66 |
| Ours | 93.33 | 89.82 | 85.00 |

Compared to the improved GMM model, the ML3D-ED model improves the mean accuracy by 13%. The method proposed in this paper shows an 8.89% improvement in accuracy compared with the best Two-Stream Inflated 3D ConvNet-Multilayer Perceptron (I3D-MLP) method. The Kappa coefficient is used for agreement testing and can also be used to measure classification precision. The calculation of the Kappa coefficient is based on the confusion matrix. The Kappa coefficient represents the proportion of error reduction produced by a model classification compared to a completely random classification. The Kappa calculation results fall within the range of $[-1, 1]$ and can be divided into five groups to represent different levels of agreement, as shown in Table 3.

**Table 3.** Meaning of Kappa values.

| Range of Kappa Values | Meaning |
|---|---|
| 0.00~0.20 | Very low agreement (slight) |
| 0.21~0.40 | General agreement (fair) |
| 0.41~0.60 | Intermediate agreement (moderate) |
| 0.61~0.80 | High agreement (substantial) |
| 0.81~1.00 | Nearly complete agreement (almost perfect) |

The Kappa value of the method in this paper is 85, which is within the range of [0.81, 1.00]. It shows an 8.34% improvement compared with the I3D-MLP method, indicating that the method proposed in this paper has a higher agreement.

Table 4 shows a comparison of the feature extraction models in this paper for the number of parameters and computational cost. We utilized the thop (https://github.com/Lyken17/pytorch-OpCounter, accessed on 13 March 2024) third-party library in PyTorch to evaluate the parameters and computational complexity of model. To ensure fairness, we adhered to the same computational resource constraints, training configurations, and uniform data processing procedures.

**Table 4.** Comparison of the feature extraction models for the number of parameters and computational cost.

| Model | Params | FLOPs |
|---|---|---|
| I3D | 12.287 M | 223.013 G |
| ML3D | 4.977 M | 49.800 G |

FLOPs represents the number of floating point operations per second, which is a standard for the computational complexity of a model. FLOP represents the number of floating point operations, and s represents seconds. The unit *G* represents a billion, denoting the magnitude of floating-point operations (FLOPs) in billions. Params represents the total parameters of the model. The unit *M* stands for a million, indicating the number of model parameters in millions.

Compared with the mainstream I3D network, the feature extraction module ML3D proposed in this paper reduces the number of parameters by 59.5% and the computational cost by 77.7%, which greatly improves the performance of feature extraction, as shown in Figure 5.

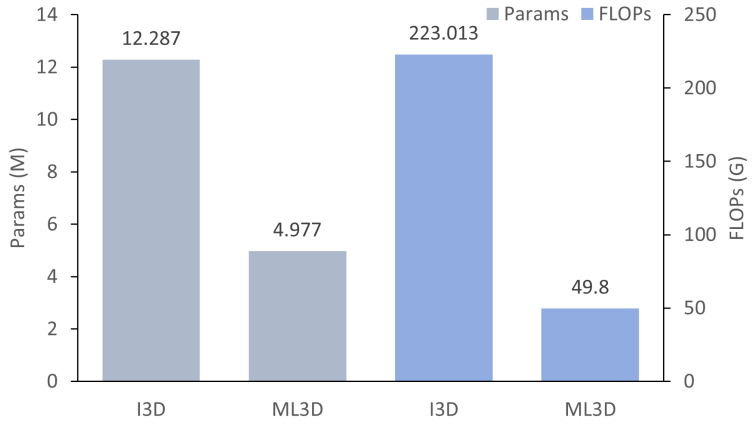


**Figure 5.** Comparison of the feature extraction models for the number of parameters and computational cost.

The runtime comparison between the I3D module and the ML3D module is shown in Figure 6. This paper uses a tensor of size (32, 3, 16, 224, 224) to simulate input data with a batch size of 32. The model is used to compute the input data $N$ times, where $N$ ranges from 50 to 1500, with intervals of 50.

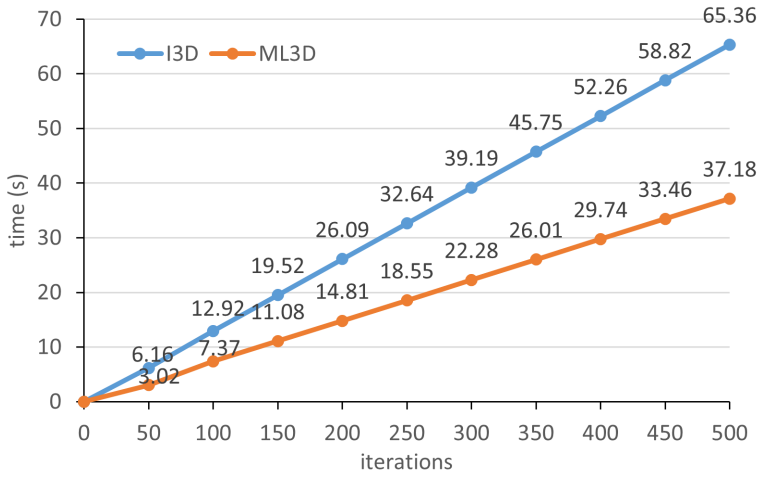


**Figure 6.** Three-dimensional convolution decoupling methods.

For 2750 iterative computations on a tensor with dimensions of (32, 3, 16, 224, 224), the I3D module takes 358.78 s, and the ML3D module takes 203.54 s. Regarding computational speed, the ML3D module shows a 43.2% improvement compared to the I3D module.

Although the calculation speed is improved and the number of parameters is reduced, the ML3D module achieves better results than the I3D module through the multi-scale feature extraction design, as shown in Table 5. We replaced the feature extraction module I3D in the most advanced I3D-MLP method with ML3D. As shown in rows 1 and 2 of Table 5, the ML3D feature extraction method has achieved an accuracy improvement of 0.83% over I3D, with an increase of 3.4% in maF1 and approximately 1% in Kappa. Meanwhile, we conducted a replacement experiment on the ED module designed in this paper. As shown in rows 3 and 4 of Table 5, the ML3D method has led to increases in accuracy, maF1, and Kappa of approximately 3.5%, 4%, and 6%, respectively. This demonstrates that the 2 + 1D decoupled architecture effectively aids the model in capturing the spatiotemporal dynamics of videos. In addition, rows 1 and 3 and rows 2 and 4 of Table 5 show that our proposed ED module has improved in three evaluation metrics

compared with MLP, especially in Kappa. The multi-scale fusion method of the ED module addresses the issue of loss accumulation.

**Table 5.** Comparison of ML3D and I3D for their performance.

| Feature Extraction | Model | Accuracy | maF1 | Kappa |
|---|---|---|---|---|
| I3D | MLP | 90.00 | 83.86 | 77.83 |
| ML3D | MLP | 90.83 | 87.16 | 79.71 |
| I3D | ED | 90.83 | 85.85 | 79.13 |
| ML3D | ED | 93.33 | 89.82 | 85.00 |

Moreover, the ED module is more lightweight than MLP, as shown in Table 6. The number of parameters that the ED module has is one order of magnitude less the MLP module, and the computational cost is two orders of magnitude less than the MLP module. This proves that the architecture of the ED module not only has higher performance, but it can also learn video features better.

**Table 6.** Comparison of ED and MLP modules for the number of parameters and computational cost.

| Model | Params | FLOPs |
|---|---|---|
| ED | 5.410 M | 5.538 k |
| MLP | 55.092 M | 689.540 k |

#### 4.3.2. Ablation Experiment Analysis

In this section, we conduct ablation experiments to analyze the contribution of the model's modules to the model's performance and the optimal combination of parameters and structures.

1. Three-dimensional convolution decoupling methods
   A 3D convolutional filter can be decoupled into a 2D convolutional filter in the spatial domain (S) and a 1D convolutional filter in the temporal domain (T). Inspired by [30], there are three combination patterns based on the interactions between two convolutional filters. The first pattern is a cascade combination of a spatial 2D filter and a temporal 1D filter. These two filters can directly interact with each other on the same path, and only the temporal 1D filter directly affects the final output, as shown in Figure 7 ML3D-a. The second pattern is a parallel combination of two filters, where each filter indirectly interacts with each other on different paths in the network, as shown in Figure 7 ML3D-b. The third pattern is a variant of the first pattern, establishing a residual connection between S and T, so that the output of S can also directly affect the output result, as shown in Figure 7 ML3D-c.

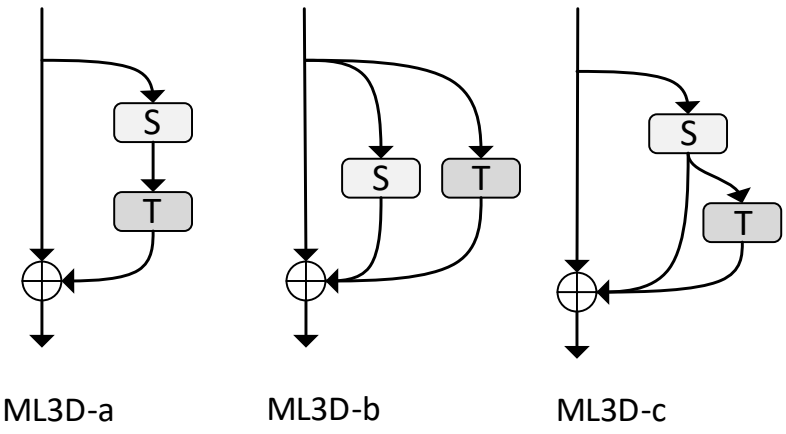


**Figure 7.** Three-dimensional convolution decoupling methods.

The effects of different decoupling methods are shown in Table 7. Different decoupling methods have a significant impact on performance.

**Table 7.** Performance of different decoupling methods.

| Decoupling Method | Accuracy | maF1 | Kappa |
|---|---|---|---|
| ML3D-a | 90.41 | 85.65 | 78.93 |
| ML3D-b | 92.08 | 87.76 | 82.18 |
| ML3D-c | 93.33 | 89.82 | 85.00 |

The fact that ML3D-b and ML3D-c perform better than ML3D-a proves that directly connecting the output of the spatial 2D filters to the final output enhances the model's information flow path, enabling spatial features to have a more direct impact on the final prediction. ML3D-c shows an improvement over ML3D-b with an approximately 1% improvement in accuracy, 2% in maF1, and 3% in Kappa, which validates that the direct influence of the two types of filters has a positive effect on the model's performance.

2. Downsampling methods

In the ML3D module, downsampling is used to reduce the feature dimensions of the raw input so that the feature dimensions are the same for residual connection. The downsampling can be designed as a 3D convolution with learnable parameters or a parameterless pooling layer for direct dimensionality reduction. Table 8 shows the number of parameters and the computational cost of different downsampling methods. Compared with parameterless pooling, 3D convolution will increase the number of model parameters by 0.005 M and the computational cost by 7.398 G. Its impact on performance is shown in Table 9. The 3D decoupling methods used are ML3D-c.

**Table 8.** Number of parameters and computational cost of different downsampling methods.

| Downsample | Params | FLOPs |
|---|---|---|
| 3D convolution | 4.977 M | 49.800 G |
| pooling | 4.972 M | 42.402 G |

**Table 9.** Performance of different downsampling methods.

| Downsample | Accuracy | maF1 | Kappa |
|---|---|---|---|
| 3D convolution | 93.33 | 89.82 | 85.00 |
| pooling | 91.25 | 85.15 | 79.69 |

The performance has been greatly improved for 3D convolution compared to pooling, particularly the Kappa value, which has been improved by 5.31%. The downsampling method used in this paper is 3D convolution, which has resulted in a trade-off of a number of parameters of 0.005 M and a computational cost of 7.398 G for a considerable performance improvement.

3. Multi-scale learning

The convolution size of mainstream I3D feature extractions is fixed, and a large amount of practice shows that capturing multi-scale information is beneficial for improving model performance. The ML3D model uses 2D convolutional filters of four scales for initial feature extraction. Table 10 shows the impact of convolutional filters of different sizes on performance.

The first row shows the performance of single-scale convolutional filters. The analysis of rows 2 and 3 of the Table shows that the combination of two small-sized and two large-sized filters has led to improvements of 1–3% across all metrics, indicating that moderately increasing filter size can enhance the model's feature extraction capabilities, thereby improving its overall performance. Compared to row 3, row 4 shows

a decrease of approximately 3% in Accuracy, 4% in maF1, and 7% in Kappa, indicating that excessively large convolutional filters have a negative impact on performance.

**Table 10.** Convolution kernels of different sizes.

| Filter Size | Accuracy | maF1 | Kappa |
|---|---|---|---|
| 7,7,7,7 | 90.83 | 86.09 | 79.75 |
| 3,7,9,11 | 92.08 | 88.46 | 82.42 |
| 3,7,13,15 | 93.33 | 89.82 | 85.00 |
| 3,7,13,17 | 90.42 | 85.22 | 77.91 |

## 5. Conclusions

In this paper, we designed an innovative ML3D, aiming to replace the traditional I3D feature extraction module. After rigorous comparison and testing, we found that, compared to the I3D model, the ML3D-ED model not only significantly reduced the number of parameters by 59.55%, but also significantly optimized the computational cost (FLOPs), with a decrease of up to 77.67%. This improvement not only significantly improved the computational efficiency of the model, but it also greatly reduced the consumption of computing resources, bringing greater convenience to practical applications.

The ML3D-ED's unique network architecture accurately captures the complex and subtle movement changes in FMS, thereby improving the accuracy of action quality assessment. Our research findings show that this method performs well in handling FMS video streams. The application and industrialization of FMS movement assessment is believed to be an interesting field for future research.

**Author Contributions:** Conceptualization, X.L. and Y.L.; methodology, C.F. and Z.C.; software, X.Y.; validation, Z.C., X.Y. and H.C.; formal analysis, X.L.; investigation, X.L.; resources, X.Y and H.C.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.Y and H.C.; visualization, X.L.; supervision, X.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China (62277010, 62302203), Science and Technology Project of Fuzhou Institute of Oceanography (2023F03), Fuzhou-Xiamen-Quanzhou National Independent Innovation Demonstration Zone Collaborative Innovation Platform Project (2022FX6), Fujian Province Education Science 14th Five-Year Plan 2022 Ollaborative Innovation Special Project (Fjxczx22-450), Fujian Institute of Education Special Research Project on Training Reform (2023PX-06) and the Fujian Provincial Health Commission Technology Plan Project (2021CXA001).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Jiao, L.; Zhang, R.; Liu, F.; Yang, S.; Hou, B.; Li, L.; Tang, X. New Generation Deep Learning for Video Object Detection: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3195–3215. [CrossRef] [PubMed]
2. Pareek, P.; Thakkar, A. A Survey on Video-Based Human Action Recognition: Recent Updates, Datasets, Challenges, and Applications. *Artif. Intell. Rev.* **2021**, *54*, 2259–2322. [CrossRef]
3. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. Sequential Deep Learning for Human Action Recognition. In Proceedings of the Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, 16 November 2011; Proceedings 2011; pp. 29–39.
4. Zhou, Y.; Sun, X.; Zha, Z.-J.; Zeng, W. Mict: Mixed 3d/2d Convolutional Tube for Human Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 449–458.
5. Spilz, A.; Munz, M. Automatic Assessment of Functional Movement Screening Exercises with Deep Learning Architectures. *Sensors* **2022**, *23*, 5. [CrossRef]
6. Duan, L. Empirical analysis on the reduction of sports injury by functional movement screening method under biological image data. *Rev. Bras. Med. Esporte* **2021**, *27*, 400–404. [CrossRef]

7. Zhou, S.K.; Greenspan, H.; Davatzikos, C.; Duncan, J.S.; van Ginneken, B.; Madabhushi, A.; Prince, J.L.; Rueckert, D.; Summers, R.M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proc. IEEE* **2021**, *109*, 820–838. [CrossRef]
8. Bertasius, G.; Wang, H.; Torresani, L. Is space-time attention all you need for video understanding? In Proceedings of the ICML, Virtual Event, 18–24 July 2021; p. 2.
9. Lin, X.; Huang, T.; Ruan, Z.; Yang, X.; Chen, Z.; Zheng, G.; Feng, C. Automatic Evaluation of Functional Movement Screening Based on Attention Mechanism and Score Distribution Prediction. *Mathematics* **2023**, *11*, 4936. [CrossRef]
10. Lin, X.; Chen, R.; Feng, C.; Chen, Z.; Yang, X.; Cui, H. Automatic Evaluation Method for Functional Movement Screening Based on a Dual-Stream Network and Feature Fusion. *Mathematics* **2024**, *12*, 1162. [CrossRef]
11. Wu, W.L.; Lee, M.H.; Hsu, H.T.; Ho, W.H.; Liang, J.M. Development of an automatic functional movement screening system with inertial measurement unit sensors. *Appl. Sci.* **2020**, *11*, 96. [CrossRef]
12. Bochniewicz, E.M.; Emmer, G.; McLeod, A.; Barth, J.; Dromerick, A.W.; Lum, P. Measuring functional arm movement after stroke using a single wrist-worn sensor and machine learning. *J. Stroke Cerebrovasc. Dis.* **2017**, *26*, 2880–2887. [CrossRef] [PubMed]
13. Hong, R.; Xing, Q.; Shen, Y.; Shen, Y. Effective Quantization Evaluation Method of Functional Movement Screening with Improved Gaussian Mixture Model. *Appl. Sci.* **2023**, *13*, 7487. [CrossRef]
14. Bai, Y.; Zhou, D.; Zhang, S.; Wang, J.; Ding, E.; Guan, Y.; Long, Y.; Wang, J. Action quality assessment with temporal parsing transformer. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 422–438.
15. Gordon, A.S. Automated video assessment of human performance. In Proceedings of the AI-ED, Washington, DC, USA, 16–19 August 1995; Volume 2.
16. Li, Y.; Chai, X.; Chen, X. Scoringnet: Learning key fragment for action quality assessment with ranking loss in skilled sports. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 149–164.
17. Tao, L.; Elhamifar, E.; Khudanpur, S.; Vidal, G.D.; Vidal, R. Sparse hidden markov models for surgical gesture classification and skill evaluation. In *Information Processing in Computer-Assisted Interventions: Third International Conference, IPCAI 2012, Pisa, Italy, 27 June 2012. Proceedings*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 3; pp. 167–177.
18. Parmar, P.; Morris, B.T. Measuring the quality of exercises. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2241–2244.
19. Xu, C.; Fu, Y.; Zhang, B.; Chen, Z.; Jiang, Y.; Xue, X. Learning to score figure skating sport videos. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 4578–4590. [CrossRef]
20. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
21. Sharma, V.; Gupta, M.; Pandey, A.; Mishra, D.; Kumar, A. A review of deep learning-based human activity recognition on benchmark video datasets. *Appl. Artif. Intell.* **2022**, *36*, 2093705. [CrossRef]
22. Hu, K.; Jin, J.; Zheng, F.; Weng , L.; Ding, Y. Overview of behavior recognition based on deep learning. *Artif. Intell. Rev.* **2023**, *56*, 1833–1865. [CrossRef]
23. Hara, K.; Kataoka, H.; Satoh, Y. Can spatiotemporal 3D cnns retrace the history of 2D cnns and imagenet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 23 June 2018.
24. Wang, X.; Miao, Z.; Zhang, R.; Hao, S. I3d-lstm: A new model for human action recognition. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *569*, 032035. [CrossRef]
25. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
27. Xing, Q.-J.; Shen, Y.Y.; Cao, R.; Zong, S.X.; Zhao, S.X.; Shen, Y.F. Functional movement screen dataset collected with two azure kinect depth sensors. *Sci. Data* **2022**, *9*, 104. [CrossRef] [PubMed]
28. Parmar, P.; Tran Morris, B. Learning to score olympic events. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 20–28.
29. Tang, Y.; Ni, Z.; Zhou, J.; Zhang, D.; Lu, J.; Wu, Y.; Zhou, J. Uncertainty-aware score distribution learning for action quality assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9839–9848.
30. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.