

A Comparative Study of Slowfast and R2plus1d Models in Basketball Action Recognition

Tianqiqi Zhu

Abstract

The main challenge in the range of basketball action recognition is that models can't effectively see the whole basketball game because of its great complexity. This paper investigates the recognition of basketball action by implementing a new kind of model slowfast. Its fast and slow pathway can complementarily give a reasonable solution to the problem. Herein, the efficacy of the proposed method is compared with the existing r2plus1d model through experiments. It turned out that the validation result of the slowfast model numerically reaches the same level as that of r2plus1d. Moreover, slowfast model can capture action details precisely, process special information effectively, and compute with low cost and high efficiency.

Keywords: basketball action recognition, slowfast, r2plus1d

1. Introduction

Recently, Action recognition is more and more commonly used in sports games, which has huge potentials. With the rising technology, the coaches can reasonably assess a player's basketball skills. In other word, seeing the accurately captured movements, the coaches can better decide the players they would like to use in a game. On the other hand, action recognition can help find some hidden problems in a player's training system. However, there is a problem for computers to see basketball games correctly, because 10 players are moving at the same time. Thus, it is considered a hard task to recognize so many movements in a short time. Many researchers have proposed various plans for improving the preexisting method such as using the pre-trained model r2plus1d[1]. Herein I used a new pre-trained model named slowfast[2], which can shed new light on the problem.

2. Literature review

Action recognition and detection is now a very important and prevailing topic in computer vision. In the range of action recognition, basketball action recognition is still a longstanding problem. The technology is still lacking and in great need. Big sports league like NBA (National Basketball Association) desperately wants a proven technique. Thus, there are many people thinking of ways to improve the accuracy of the recognition and the efficiency of training in this range. Some of them would like to change the algorithm[3] involved in action recognizing or tracking players, some of them focus on the use of datasets[4], others may put their effort in different models

(methods) like the series of YOLOs (You Only Look Once)[5] and CNN. The 3 ways are all helpful for improving the performance of the identifier.

2.1 New Algorithm Dynamic Residual Attention Mechanism

The article[3] proposed that the original C3D (Convolutional 3D) is insufficient for action recognition in the range of basketball. There are still problems such as the difficulty of focusing on keyframes. Therefore, the method of dynamic residual attention mechanism is introduced, which can provide a solution to the problem. By using this way to extract sufficient feature information, the identifier can capture the pose of the player in the filtered images. Then, the accuracy for action recognition may improve.

In the real world, basketball games are in the form of videos. However, the research mainly focuses on the images. The actions in the videos are much harder to be recognized. As a result, the improved algorithm is not as useful for the case in the real world. In fact, algorithm is usually very hard to make a change.

2.2 Change of Models and Methods

Models is a very pivotal part in the process of recognizing player actions in a basketball game. There are a variety of models to choose from in recognizing the players' action. Different models can result in different result, they have various focuses, some focus on the efficiency of the process, others may focus on the accuracy of the identification. Such models include YOLO and CNN. The following papers all

dive deep into the ways to improve recognition of player actions, but they don't have the same amounts of actions, some have 8 moves, some have less.

2.1.1. Skeletal Nodes

Tianyu Fan discovers the unique use of skeletal nodes[5]. With the use of nodes, YOLOv8 and the network structure named STGCN (Spatio-Temporal Graph Convolutional Networks) as the core, the study offers a new way to improve the existing solution, which allows the recognition to be more accurate.

2.1.2. Bidirectional Long Short - Term Memory

BiLSTM (Bidirectional Long Short - Term Memory)[6] model is used in another research. They have a small amount of the datasets and does research on ways of action recognition for multiple players, with videos of different level action recognition. Eventually, they propose the solution with 80.6% accuracy.

2.1.3. CNN and LSTM

CNN (Convolutional Neural Network) and LSTM (Long Short - Term Memory) are being well trained using three datasets "NTURGB+D", "Basketball-Action Dataset", and "B3D Dataset" in the paper [4], and the datasets are being processed. The method offers a novel insight for the way to recognize player action in a video. On the other side, the article also points out some severe problems, like insufficient utilization of the specific characteristics and requirements of basketball action recognition. Besides, there are hardly any evaluation metrics. These are the problems that need to be noticed.

2.1.4. CapsNets and ARPO

Another model is used to improve the accuracy of the recognition in the research [7], the model is a combination of the CapsNets or Capsule Neural Networks with an ARPO or augmented variant of Red Panda Optimizer. The research typically solves the issue of the complexity of multiplayer in a basketball game. The paper's objective is to achieve precise and robust basketball action recognition along with the analysis through the application of DL techniques.

2.1.5. Slowfast network

Lin Zhang creatively use the slowfast[8] model as a network to train other models like YOLOv4, it gives a special insight for the dual structure of the model, which hints me the use of the fast and slow part of the different models.

2.1.6. real-time multi-person detection and tracking framework

David Acuna[9] mainly focuses on player tracking, he uses a real-time multi-person detection and tracking framework based on YOLOv2, which can perform detection and tracking of basketball players on sequences of videos. He also makes use of the dataset "NCAA Basketball Dataset".

In all, the dataset for basketball is still limited, we still need much more datasets to further improve the accuracy of the identifier, we already try making change in the algorithm, and models. There are huge number of models for us to choose, which can bring distinct benefits for the field. With the use of multiple models, we can see which model have the best result by looking at the papers.

3. Methodology

An open-source project is used in this paper[10]. The project uses r2plus1d pre-trained model along with 3D CNN architecture.

3.1 The Dataset

The dataset I chose is SpaceJam[11] that is provided online, it consists of both the clips and joints. The identifier of "parallel" examples will be the same, while what distinguishes them is the extension of the files on which they are stored. The first part of the dataset called clips is made up from 16 frames in RGB focused on one player, while each of the second part of the dataset called joints includes the coordinates (x, y) on the image plane of the player's joints.

The joint stores the following information:

- The x coordinate of the joint (x).
- The y coordinate of the joint (y).
- The identifier of the joint type (type).
- The identifier of the person in the frame to which that joint belongs (person ID).
- The name of the joint associated with its type identifier (name).

The joint serves as a starting point for the clip, the clip further uses the x and y to derive the top left and bottom left coordinates of the video. The formula is shown below. Given a pose P formed by N joints $[g_1, g_2, \dots, g_n]$, the bounding-box must be enlarged by a certain percentage (w, h) with respect to x and with respect to y.

$$X_{top\ left} = \min \{g_i (x)\} - \Delta w \times \min\{g_i(x)\}$$

$$Y_{top\ left} = \min \{g_i (y)\} - \Delta h \times \min\{g_i(y)\}$$

$$X_{bottom\ left} = \max\{g_i(x)\} + \Delta w \times \max\{g_i(x)\}$$

$$Y_{bottom\ left} = \max\{g_i(y)\} + \Delta h \times \max\{g_i(y)\}$$

During the creation of the dataset, some anomalies were found in the part of the joints such as the wrong recognition of different key points representing knots, which may mislead the identifier. Eliminating anomalous examples can be done to solve this problem.

The dataset contains 10 classes including walk, no action, run, defense, dribble, ball in hand, pass, block, pick, shoot for the identifier to recognize. The number of examples[10] is shown for every class in Figure 1. The total size of the final annotated datasets is about 32560 examples, which should be enough for the model to recognize different actions. However, the examples are further disposed, because some classes like pick, block, pass have not as often appeared compared to walk, no action in a basketball game. Then, I can't evenly collect the same amount of the actions in a game. In this case, the examples for these classes are not evenly distributed, which may lead to some problems for the identifier to recognize these certain types of classes. As a result, the dataset is further augmented, the short videos are rotated and translated. There will be more videos for the model to train. After augmentation, the dataset has more than 49901 examples.

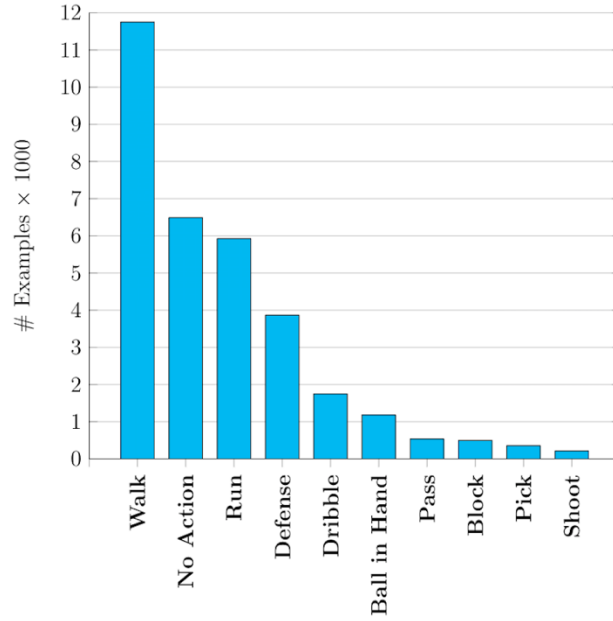


Figure 1 Histogram Number of Examples per Class

3.2 Tracking the Players

Since we are living in a 3D world, it is not as easily for the identifier to recognize a player. Thus, to assign a label to each pose of each frame for some consecutive time, a method called nearest neighbor[11] is used as shown in Figure 2. By calculating the total number of distances between different moves. A_x , A_y refers to the x and y coordinates of the original image corresponding to a pixel in the target image, while B_x , B_y represents the x and y coordinates of a pixel of the target image. A_w and A_h are the width and the height of the original picture. B_w and B_h are the width and height of the target picture. We can assign the same identifier between two poses that have the minimum distance.

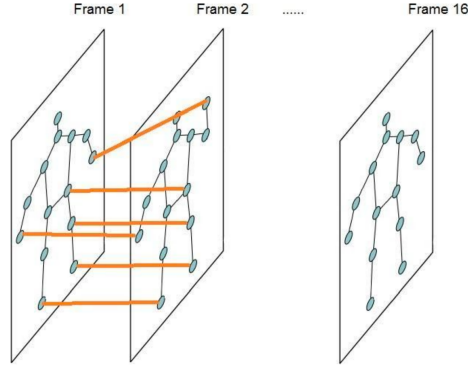


Figure 2 Position Tracking

$$Ax = Bx \times (Aw \div Bw)$$

$$Ay = By \times (Ah \div Bh)$$

3.3 R2plus1d and Slowfast Model

The original model is a pretrained baseline r2plus1d CNN (pretrained on kinetics-400 dataset). It is a model from torchvision. The 3D convolution is decomposed into a 2D spatial convolution and a 1D temporal convolution. This decomposition enables the network to learn features separately in the spatial and temporal dimensions, enhancing the model's ability to model spatio-temporal information. Each r2plus1d convolution block contains a specific number of 2D and 1D convolution filters. By reasonably setting hyperparameters, the number of parameters in the convolution block is made equivalent to that of a full 3D convolution.

At the convolution 1 layer, r2plus1d decomposes it into 2D filters and 1D filters:

$$P_{3D} = N_{3D} \times T \times D^2$$

$$P_{2D} = M \times 1 \times D^2 = M \times D^2$$

$$P_{1D} = N_{1D} \times T \times 1 \times 1 = N_{1D} \times T$$

$$P_{total} = P_{2D} + P_{1D} \approx P_{3D}$$

For each 3D or 2D or 1D filter, the number of parameters is the volume of the filter, the product of the time dimension is T, and the space dimension is D. Ideally, for the decomposed convolution operation to be computationally like the previous 3D convolution operation, we would like the total number of resolved parameters P_{total} to be approximately equal to the number of parameters P_{3D} of the original 3D convolution.

A nonlinear ReLU (Rectified Linear Unit), $f(x) = \max(0, x)$, can increase the model's nonlinear expression ability, performing outstandingly on multiple action recognition benchmark datasets.

Let x be the size of an input clip of size $3 \times L \times H \times W$. L means the number of clips in the frame, W and H are the width and length of a short video. Make z_i be the output tensor computed by the i -th convolutional block in the residual network. Each block consisting of two convolutional layers with a ReLU activation function after each layer.

The output of the i -th residual block is shown as follows

$$Z_i = Z_{i-1} + F(Z_{i-1}; \theta_i)$$

$F(\cdot; \theta_i)$ implements the composition of two convolutions parameterized by weights θ_i and the application of the ReLU functions.

Herein, slowfast is chosen to replace the original neural network model for video recognition. It captures spatial semantics and motion information in videos through two

pathways, namely the slow pathway and the fast pathway, achieving excellent performance in action classification and detection tasks, which is extremely suitable for basketball recognition. slowfast's advantages in basketball action recognition mainly stem from its unique architecture design, which can effectively process complex information in basketball videos.

Dual-Pathway Architecture[2] (shown in Figure 3) is unique. Slowfast composes of a Slow pathway and a Fast pathway. The Slow pathway works at a low frame rate and is used to capture spatial semantic information, good at identifying features such as object categories, colors, and textures in the video. Slowfast has a large temporal stride and processes fewer frames. On the other hand, the Fast pathway operates at a high frame rate, focusing on capturing motion information with fine temporal resolution. Although it has a lower channel capacity, it can quickly sense changes in actions. As a result, the two pathway is complementary.

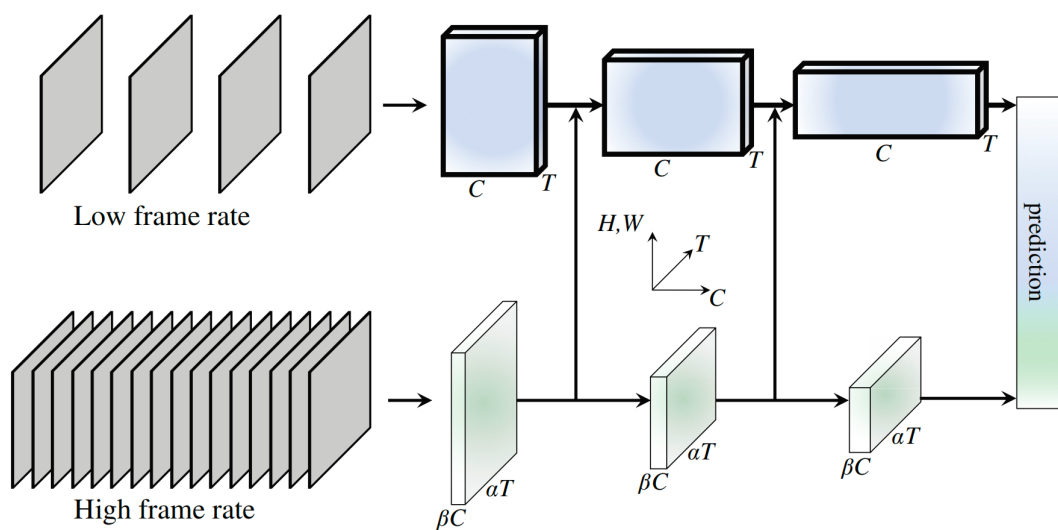


Figure 3 Slow and Fast Pathway of Slowfast

The Fast pathway achieves lightness by reducing its channel capacity. For example, the number of channels is $1/8$ of that of the Slow pathway, and its computation only accounts for about 20% of the total computation. This allows slowfast to maintain high temporal resolution without imposing excessive computational burdens. The weak spatial modeling ability of fast pathway emphasizes the capture of temporal information.

Let's be specific on the range of basketball, slowfast network can precisely capture action details, which is a huge advantage. Basketball games are characterized by fast changing actions. For example, players' shooting, passing, dribbling, and defensive actions have complex posture changes in a short time. The fast pathway of the slowfast model, operating at a high frame rate, can precisely capture the details and temporal dynamics of these fast actions, accurately identify the start, process, and end of actions, and improve the accuracy of action recognition. On the other hand, other models can't have the ability to view every details carefully.

In a basketball scene, spatial elements such as players, the basketball, and the court are complex and essential. The slow pathway can focus on analyzing these spatial semantics, identifying player identities, court layouts, and other information. Combining with the motion information from the fast pathway, it can better understand the scene and context in which basketball actions occur. For example, it can determine the relationship between a player's position on the court and their actions, enhancing the understanding of the meaning of the actions. Slowfast is effectively in processing spatial information, which may be helpful for the recognition of players action and

future research in the domain of sports and especially basketball.

Lastly, the lightweight design of the Fast pathway enables the model to have a low computational cost when processing basketball videos. Since its computation only accounts for about 20% of the total computation, it can quickly process a large amount of video data while ensuring recognition accuracy, meeting real time requirements. It can also rapidly identify various actions in a basketball game, providing timely support for applications such as game analysis and referee assistance, which is much more efficient.

But there is a question in the process. When making use of the model, the slowfast model can be found both in its official library, and in the Pytorchvideo library. The one in Pytorchvideo library is pre-trained on kinetic 600 dataset. This dataset includes 600 classes, each class contain 600 sample videos, all with around 10 seconds. This dataset contains sports videos, which provide an abundant data sample for training, and can be helpful for the recognition tasks. The slowfast from the official library[12] is being trained fully by the SpaceJam dataset from the start. In this case, I tried models from two libraries to compare the r2plus1d training structure with the same dataset “SpaceJam”.

4. Results

4.1 Comparing Between R2plus1d and slowfast

I have already tested the model r2plus1d, and both the slowfast model (one in the library of Pytorchvideo, and the other is in its official library). The results are as

follows.

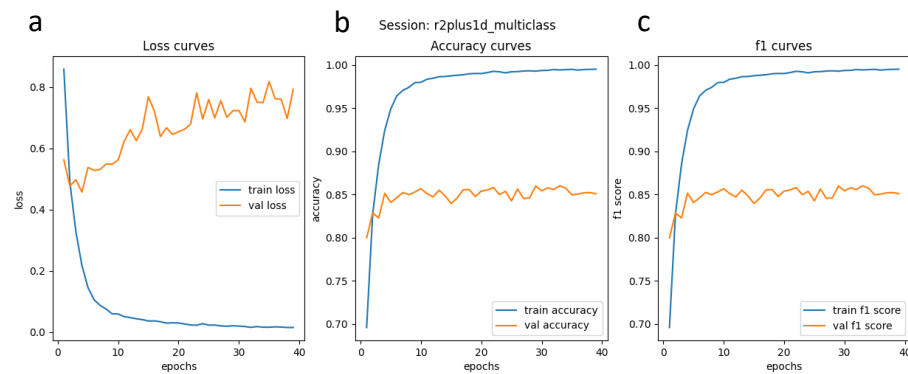


Figure 4 R2plus1d Result

This is the result of the r2plus1d model. Picture a in Figure 4, which can let us see the loss curves, is the result of training loss and the validation loss. As we can see, in 40 epochs, the training loss forms a perfect curve from more than 0.8 to almost 0, while the validation loss is unstable, it has more twists and turns, which may result from the randomness of the augmentation of the dataset, since it has a 50% of the probability of being augmented. What's more, one other reason may contribute to the result is that during model initialization, the parameters of some layers are frozen. To be specific, only the parameter of the last three layer is renewed. In other words, the layers beside layer 3, layer 4, and layer "fc" (where 10 classes contain) is not changed, resulting in the models not being able to adequately learn data features.

Picture b and c in Figure 4 can show the performance of the model. The result is almost the same, since accuracy and f1 is bounded together. The validation accuracy remains constantly around 85% after about 5 epochs, which seems to be a satisfying result.

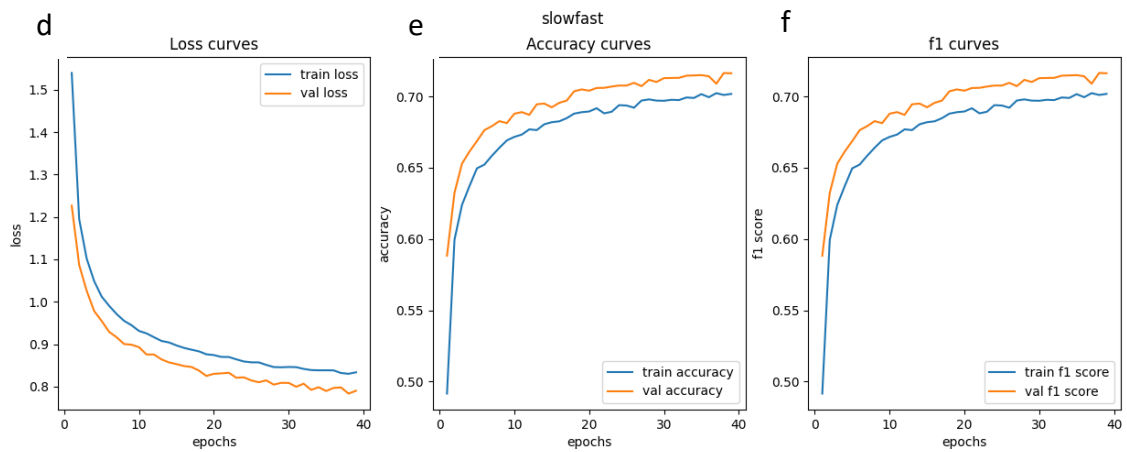


Figure 5 Slowfast (Pytorchvideo) Result

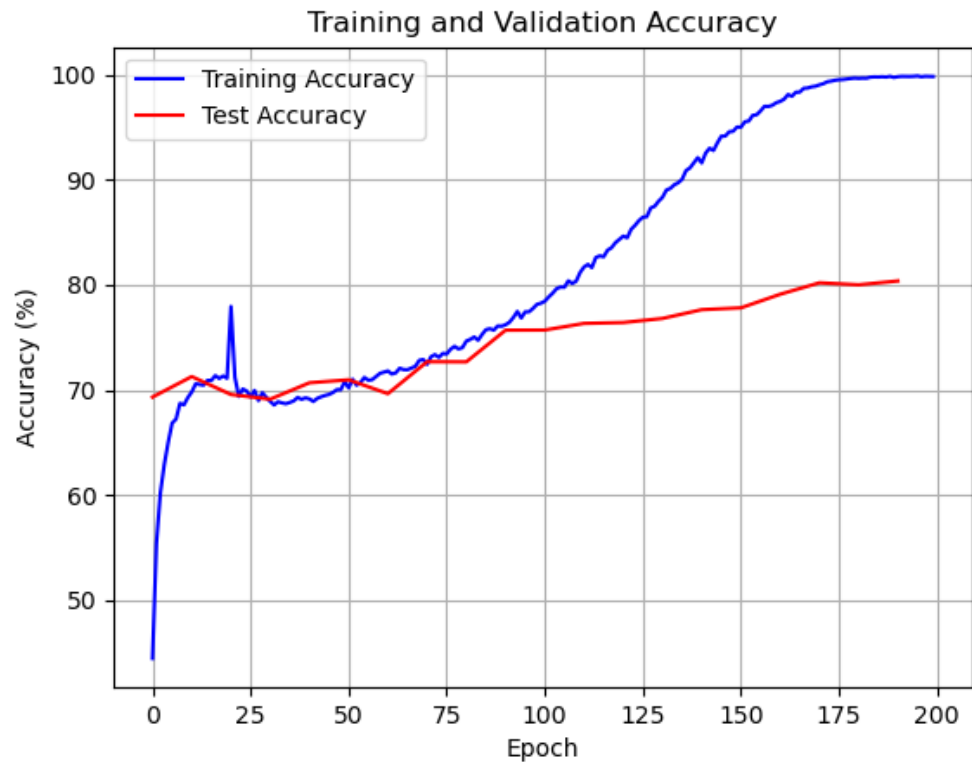


Figure 6 Slowfast (official library) Result

These are results of the alternative model slowfast. Picture d in Figure 5 is the result of the slowfast model from Pytorchvideo library. As we can see the slowfast model has a relatively smooth curve in the loss. And the training and testing accuracy of the model is very low shown in picture e and f, all at about the accuracy of 70%. I tried 80 epoch,

but the results all seem to wander around 70%.

On the contrary, I tried the slowfast model from its official library. The training and testing accuracy are shown in Figure 6. The slowfast model got an accuracy of about 80% in the long run of 200 epochs.

Each epoch, r2plus1d spends about 17 minutes but slowfast takes only 12 minutes. Thus, slowfast can train faster than r2plus1d. Note that one Nvidia 4080 super is used.

4.2 Comparing between Pytorchvideo slowfast and official slowfast

An interesting point appeared. It also seems that the slowfast model from its official library has a better performance compared to the one from the library of the Pytorchvideo, which is weird since the model is the same, the dataset is the same. why can there be a difference in the results?

The reasons may lie in three parts. First is the optimizer configuration differences, the Pytorchvideo slowfast model uses the Adam optimizer. The official slowfast model used SGD plus Momentum optimizer, which is more suitable for fine-tuning scenarios. The difference of the optimizer can lead to unstable convergence. Thus, the accuracy is different.

Secondly, the learning rate difference can also make the two results different. The Pytorchvideo model has a stable learning rate of 0.0001, the other model from the official library makes use of the dynamic learning rate scheduling (such as warmup plus cosine attenuation). In other words, the learning rate can be changed based on the situation, which is a more reasonable. Fixed learning rate is hard for balancing

parameter updates at distinct layers.

What's more, the parameter freezing policies are different. The model from Pytorchvideo only trains the last layer of the model (i.e., blocks.6.proj), while the official slowfast model will fine-tune all layers, so that the official slowfast can better deal with new tasks such as basketball action recognition. As the result, the Pytorchvideo slowfast model can be underfitted which can take less advantage of pre-trained features.

5. Discussion

This paper has generated several results that can satisfy the original aim, which is improving the preexisting method for basketball action recognition. An alternative model slowfast has provided valuable insights into the difference in the realization of training models. As we proceed with the discussion, we will dive deep into these findings, examining the significance of slowfast, and how slowfast compares with existing models such as r2plus1d. The result showed the great importance of the hyperparameters, which can directly influence the performance of a model. It points out that if all the layers of the model can be roughly changed when training, the testing performance can be better.

6. Conclusion

With the unique structure of the slowfast network, the slowfast model showed its great ability to view complex videos which is suitable for action recognition in sports

particularly for basketball. Compared with r2plus1d, slowfast spends less time for each epoch. The results of different slowfast models in both the Pytorchvideo library and its official library are compared and analyzed. The optimizer, learning rate and change of the number of layers can all contribute to the existing result. There also have some limits in my research. Only 10 actions are recognized, while more actions are needed to analyze in a real basketball game. In the future, based on the previous ways to recognize a player's action, researchers can focus on how to relate the action recognizing skill to statistics. It is a necessary job since audiences can easily know whether a player plays well in a game or in a period of time by viewing the calculated numerical value such as RAPM (Regularized Adjusted Plus-Minus) proposed in [13]

References

1. Du Tran, H.W., Lorenzo Torresani,2, Jamie Ray, Yann LeCun, Manohar Paluri, *A Closer Look at Spatiotemporal Convolutions for Action Recognition*. 2018.
2. Christoph Feichtenhofer, H.F., Jitendra Malik, Kaiming He, *SlowFast Networks for Video Recognition*. 2019: p. 10.
3. Xiao, J., W. Tian, and L. Ding, *Basketball Action Recognition Method of Deep Neural Network Based on Dynamic Residual Attention Mechanism*. Information, 2022. **14**(1).
4. Wang, J., L. Zuo, and C. Cordente Martinez, *Basketball technique action recognition using 3D convolutional neural networks*. Sci Rep, 2024. **14**(1): p. 13156.
5. FAN, T., *Real-Time Basketball Shooting Action Recognition*. 2024.
6. ZAKHARCHENKO, I., *Basketball Pose-based Action Recognition*. 2020.
7. Sha, N., *Basketball technical action recognition based on a combination of capsule neural network and augmented red panda optimizer*. Egyptian Informatics Journal, 2025. **29**.
8. Zhang, L., *Deep learning based fine-grained recognition technology for basketball movements*. Systems and Soft Computing, 2024. **6**.
9. Acuna, D., *Towards Real-Time Detection and Tracking of Basketball Players using Deep Neural Networks*. 2017.
10. hobin. 2021; Available from: <https://github.com/hkair/Basketball-Action-Recognition>.
11. Francia, S., *Classification of Basketball Actions through Deep Learning Techniques*. 2017.
12. Facebook_Research. Available from: <https://github.com/facebookresearch/SlowFast>.
13. Byman, M., *BUILDING A STATISTICAL LEARNING MODEL FOR EVALUATION OF NBA PLAYERS USING PLAYER TRACKING DATA*. 2023.