TSN vs I3D vs Pose-C3D: Action Recognition in Basketball using SpaceJam Dataset

Tafadzwa Blessing Chiura
Academy of Computer Science and Software Engineering, Faculty of Science
University of Johannesburg
Johannesburg, Gauteng, South Africa
bless.mukuru@gmail.com

Dustin van der Haar*
Academy of Computer Science and Software Engineering, Faculty of Science
University of Johannesburg
Johannesburg, Gauteng, South Africa
dvanderhaar@uj.ac.za

Abstract

Most of the research conducted in action recognition is mainly focused on general human action recognition, and most of the available datasets support studies in general human action recognition. In more specific contexts, such as basketball, datasets that are as comprehensive and publicly available are limited. This study proposes taking three popular and mature methods in the field of action recognition, namely Temporal Segment Networks (TSN), Two-Stream CNN using Inflated 3D-convolutional Neural Networks (I3D) and Pose-C3D, and applying them to the SpaceJam dataset, which is a basketball-specific action dataset. All three experiments used pre-trained ImageNet models and were fine-tuned on the SpaceJam dataset. TSN was the oldest of the methods but obtained the best results of the three experiments, scoring a top-1 and top-5 accuracy of 59% and 96%, respectively. I3D was second best, with a top-1 and top-5 accuracy of 41% and 85%, respectively. Pose-C3D came in third, scoring a top-1 and top-5 accuracy of 15% and 50%, respectively. The results show that the models cannot distinguish significantly between some actions, such as ball in hand, pass and dribble. The study shows that it is feasible for context-specific fine-grain action recognition, but more needs to be done to discriminate against similar actions.

CCS Concepts

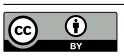
• **Computing methodologies**; • **Artificial intelligence**; • **Computer vision**; • **Computer vision tasks**; • **Activity recognition and understanding**;

Keywords

Action Recognition, Basketball, Computer Vision

ACM Reference Format:
Tafadzwa Blessing Chiura and Dustin van der Haar. 2024. TSN vs I3D vs Pose-C3D: Action Recognition in Basketball using SpaceJam Dataset. In *2024 The 7th International Conference on Computational Intelligence and Intelligent Systems (CIIS) (CIIS 2024), November 22–24, 2024, Nagoya, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3708778.3708788

*Corresponding author.



CIIS 2024, Nagoya, Japan

ACM ISBN 979-8-4007-1743-7/24/11
https://doi.org/10.1145/3708778.3708788

1 Introduction

Action recognition is a research field that continues to grow in popularity, particularly because of the vast number of use cases, such as surveillance, motion tracking and event detection [1] [2]. General action recognition has steadily improved its performance over recent years. The improvements can be attributed to the advancement in computer hardware, the growing number of publicly available datasets and innovative approaches from the wider computer vision fraternity [2]. However, more research is still required in specific contexts such as basketball, partly due to the need for comprehensive datasets and partly due to the complexity of the actions involved.

This study explores the performance of some mature and previously popular action recognition methods on a basketball-specific dataset, i.e. SpaceJam [3]. In addition, the study also looks to build on the work done by Chiura and van der Haar by providing comprehensive results on the approaches' performances in relation to previous publications and providing enriched analysis by way of a confusion matrix(s) for the three approaches, and ablation and a Neighbourhood Components Analysis (NCA) for the best-performing approach [4]. The approaches used are Temporal Segment Network (TSN), 2-stream Convolutional Neural Network with Inflated 3D CNN (I3D) and Pose-estimation (Pose-C3D).

The remainder of the paper is structured as follows: the next section will discuss some related works that have been carried out and some of the popular datasets used in the field. Then, the section discusses the background of the incorporated methods. The proceeding section will outline the experiment, data setup, and results obtained. The paper will then be closed off by a conclusion.

2 Problem Background

Sports and technology have intertwined over time as technology advances and breaks barriers. In sports, the use of technology includes tasks such as (but is not limited to) score prediction, performance tracking, instant replays and player training [5] [6]. Action recognition falls under Computer vision (CV); traditionally, action recognition has two main groupings, which are human-to-human and human-to-object interactions [7]. The tasks involved in action recognition are mainly comprised of locating the subject of interest in one frame and then keeping track of said subject in the following frames for the duration of the action and/or the video. Host and Ivašić-Kos define actions in sports as a supervised learning task on sports movement [8]. Within a game of basketball, many complex actions are performed by the players repeatedly. These

actions include running, jumping, shooting, dribbling, and passing a basketball [9].

2.1 Datasets

Creating a comprehensive dataset from scratch has proven to be time-consuming and may, at times, be financially expensive (e.g., equipment and labour costs) [10]. It is also beneficial to the community at large when common publicly accessible datasets are used as it is easier to set benchmarks and to determine the best-performing approaches. Some popular datasets are UCF-101, Sport-1M and Youtube-8M.

2.1.1 UCF-101. UCF-101 is a dataset from the University of Central Florida with 101 common action classes spanning 13000 videos and 27 hours. These actions range from applying lipstick to boxing a punching bag to cliff diving, shooting, and dunking a basketball [1] . At publication, the authors described UCF-101 as "the largest dataset of human actions". The video clips are all sampled at 25 fps (frames per second) with a 320 x 240 pixels resolution. The average length of each clip is 7.21 seconds, the shortest clip is over 1 second, and the longest lasts 71 seconds. Before UCF-101 was published, the action recognition community faced two main issues with most of the popular datasets; the first was that of scale and size. The dataset is very limited compared to the vast number of human actions. The second issue was that most videos were captured and/or recorded in very controlled environments that sometimes failed to match real-world actions and movements.

2.1.2 Sport-1M.. Sports-1M, as the name implies, is a dataset made up of 1 million videos that span over 487 different sports activities [11]. This dataset was introduced to show the effectiveness of CNN architecture on feature learning. The activities or actions are specific in nature and do not follow the convention of general "actions", as seen in the above dataset(s). The action classes were manually arranged into major action classes, namely, aquatic sports, team sports, winter sports, ball sports, combat sports and sports with animals. To elaborate on the fine-grain aspect of the Sport-1M, the dataset contains six different types of bowling, 23 types of billiards, 3-on-3 and wheelchair basketball, and swimming, both medley and synchronised swimming. The videos are, on average, 5 minutes and 36 seconds long. There are between 1000 to 3000 videos per class, and about 5% of the dataset only has multiple action labels.

2.1.3 Youtube-8M.. YouTube-8M takes large-scale datasets to another level. It consists of 8 million videos sourced from the popular video-sharing platform YouTube [10]. The videos combine for over half a million hours and are spread across 4800 classes, each with three labels on average. The average video duration is 230 seconds, the complete dataset is over 1.9 billion frames, and the videos were sampled at one frame per second. This dataset further preserves the audio modality, which has the notable merit of added context. For example, an interview with Kobe Bryant can be classified under basketball. Google conducted this project to address the issues of storage and computation in the computer vision community. Youtube-8M was motivated by the improved results in image understanding fields, which were mostly accredited to the availability of large-scale image datasets such as ImageNet [12].

ImageNet, in summary, is a large general image dataset with diverse human and object subjects in the images. Youtube-8M set out to do the equivalent in the video realm, providing a "large-scale dataset for general multi-label classification". Youtube-8M labels go beyond just action classes as other popular datasets do, such as Sports-1M and UCF101. The annotations also consider objects, scenes, and events in the videos, which allows the dataset to cater to games, sports, hobbies, and other human-related activities. The objective of the dataset and its labelling strategy is to go beyond understanding what is happening in the video at a frame-by-frame level and to be able to pick out the key points that would help best describe what is being carried out in said video.

2.1.4 SpaceJam Dataset. SpaceJam is a relatively small action recognition dataset compared to other popular datasets. SpaceJam is an action recognition dataset specifically geared towards basketball [3]. The dataset spans ten action classes spread over some 32000 short video clips. Each video clip has up to 16 frames and lasts 2 to 4 seconds. The actions covered include running, shooting, passing and dribbling. It is worth noting the uneven distribution of the action classes in SpaceJam, e.g. the action class with the most samples has almost 12000 samples/clips, whilst the action class with the least number of samples/clips has 426 (see Figure 1). Another drawback of SpaceJam is that a significant portion of the dataset is actions that are not specific to basketball, i.e. more than half the samples are either "no action", "walk", or "run", which can be performed outside of the basketball domain. The skewed distribution of actions and the fact that some key basketball actions, such as "pick and roll" and dunking, are missing limit the study.

However, SpaceJam is the selected dataset for this research work. One of the merits of the dataset is that all video clips were captured from one camera angle (a raised position in the centre of the court) to mitigate problems such as motion blur. The other merit is that the video capture is dynamic and thus limits sudden movements as it focuses on the area where the action is taking place, making for more harmonic and smooth videos. Finally, the dataset provides multiple modalities, allowing a wider range of methods and experiments to be carried out; the modalities are skeleton pose information and RGB video clips. RGB imagery is a visual depiction made up of three colour combinations: red, green and blue [13].

The role that datasets play in the field of action recognition cannot be overstated. Datasets go hand-in-hand with action recognition methods as they support the advancement of the field. It is also seldom that a discussion be conducted about one without the other, and following that convention, the next section will look at some methods that have been incorporated.

2.2 Similar Work

Pan and Li focused on solving player tracking in basketball footage [14]. They primarily focused on upper body movements to decipher motion, as these were proven more distinct to ensure effective recognition. The upper body parts were deemed the most distinct to ensure effective recognition. The motion features were extracted from the frames using the Lucas-Kanada algorithm. The motion blocks are constructed by selecting connected frames and are fed into a bi-directional recurrent neural network.

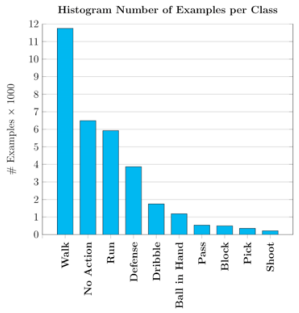


Figure 1: SpaceJam dataset action class distribution. Image from Francia (https://github.com/simonefrancia/SpaceJam (2015)

Liu's approach involved motion recognition and tracking in basketball using pose estimation to detect human body key points and a long short-term memory (LSTM) based model as the feature extractor [15]. The researcher(s) aimed to enhance training and reduce injuries using sensors to track movements. Their training set was not limited to basketball; it included players from badminton, soccer, and gymnastics. Their approach achieved an average recognition rate greater than 70%. It is worth noting that the initial LSTM that was applied led to large information losses; the information loss was because of how LSTMs prioritise the most recent data and drop the historical information. The remedy to the information loss was rather to apply a bi-directional-LSTM.

Nistala and Guttag designed a system to track offensive plays in basketball [16]. The objective was to gather, map and examine player movement patterns when playing offence. The system used a learning network that leveraged an autoencoder architecture. After the encoding/decoding phase, the output is stored as tuples of information containing trajectory embedding as one field. The said field lists the X and Y-coordinates of the player's position per frame. Afterwards, k-means clustering is applied to find groupings of similar movement patterns to produce the final outputs.

3 Methods and Experiment Setup

Several methods and approaches have been proposed and implemented to tackle action recognition, and these approaches involve using convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM), and deep learning. The following sections will examine the methods selected for this study. To ensure a like-for-like comparison, all the methods used models pre-trained on Kinetics-400 and a ResNet50 backbone. Each experiment was trained using Google's Colab platform and was allocated 20 epochs each, as this was about as much as could be done on the free tier and no early stopping incorporated.

Further, to ensure that every action class had fair weighting on the model(s) training, given the skewed distribution of samples, part of the pre-processing step involves extracting an equal number of videos per class, also referred to as stratified sampling. The data partitioning into training and validation sets will follow the conventional 70:30 rule. The shoot action, which has the least samples, is made up of 426 videos, which leads to the training set being made up of 298 videos from each action class and 127 videos for the evaluation set. The final dataset resulted in 2980 training videos and 1270 validation videos.

3.1 Temporal Segment Networks (TSNs)

TSNs are known to be effective and lightweight methods suitable for understanding time series data. Part of what makes TSNs attractive is that they are not concerned with long-duration data and work well with short and constrained training data [17]. Temporal segment networks build on 2-stream CNNs, which separate the learning into spatial and temporal streams. TSN uses RGB difference for the spatial stream and warped optical flow fields for the temporal stream. Table 1 shows the results published for TSN on Kinetics-400 and Something-Something.

These parameters are similar to those used by Wang et al. [17]. SGD (stochastic gradient descent) is applied as the loss optimiser for the network. The batch size was set to 8, with a dropout ratio of 0.4 and momentum of 0.9, and the input frames were resized to 224 x 224.

3.2 2-Stream inflated 3D-CNN (I3D)

I3D is drawn from 2D-CNNs used for image classification and expanded into 3D; the expansion makes it possible to learn spatiotemporal features extracted from videos [18]. To bootstrap parameters from pre-trained ImageNet models, an image can be turned into a "boring video" by copying the image repeatedly into a video sequence. The 3D models can then be implicitly pre-trained on ImageNet by satisfying the boring-video fixed point. The boring-video fixed point can be realised thanks to linearity by replicating the weights of the 2D filters N-times along the time dimension and rescaling them by dividing by N. Ensuring that the convolutional filter response is the same. Table 2 shows the results published for I3D on Kinetics-400 and Something-Something.

I3D takes input in the form of frames, with each video represented by a directory containing 16 frames, as in the case of SpaceJam. I3D also uses an SGD optimiser; batch normalisation is applied at each layer, momentum is set to 0.9 ReLU (rectified linear unit) as the activation function, centre cropping is used with 224 x 224 dimensions and a stride of 2. These parameters are similar to those used by Carreira and Zisserman [18].

3.3 Pose-C3D

Pose-C3D is a form of skeleton estimate that uses CNNs and RNNs to predict the 3D positions of human joints from 2D images [19]. Pose-C3D can handle the following aspects: varying lighting conditions, occlusions, and diverse human body pose. PoseC3D is applied using estimated 3D human poses as input to a classification model. The extraction of features is done using poses, and the features are used to train an action classifier. SpaceJam includes extracted joint data and video clips, making it possible to explore Pose-C3D along with the other methods mentioned above. Table 3 shows the results that were published for Pose-C3D on Kinetics-400.

Pose-C3D relies on a different modality, as discussed in the previous section. The architecture includes a 2D-top-down pose extractor, Faster-RCNN detector and RGB inputs. The setup included

Table 1: Top-1 and Top-5 accuracies of Kinetics-400 and Something-Something using TSN.

Dataset	Top-1 Accuracy	Top-5 Accuracy
Kinetics-400	74.12	91.34
Something-Something	35.51	67.09

Table 2: Top-1 and Top-5 accuracies of Kinetics-400 and Something-Something using I3D.

Dataset	Top-1 Accuracy	Top-5 Accuracy
Kinetics-400	74.80	92.07
Something-Something	48.60	77.90

Table 3: Top-1 and Top-5 accuracies of Kinetics-400 and Something-Something using Pose-C3D.

Dataset	Top-1 Accuracy	Top-5 Accuracy
Kinetics-400	47.70	-
Something-Something	-	-

Table 4: Top-1 and Top-5 accuracies of experiments carried out using SpaceJam Dataset.

Experiment	Top-1 Accuracy	Top-5 Accuracy
TSN	59.92	96.46
I3D	41.34	85.91
Pose-C3D	15.71	50.36

an SGD optimiser, the loss function is cross-entropy, momentum set to 0.9, a learning rate set at 0.2 and a batch size of 32. This configuration aligns with what Duan et al. used [19].

4 Experiment OutComes

The results for TSN, I3D and Pose-C3D on SpaceJam are shown in Table 4. TSN yielded the best results from the group, as shown in Table 4. The results for TSN are comparable with those attained on other datasets (Kinetics-400 and Something-Something), shown in Table 1, Carreira and Zisserman [18]. These results show promising signs, mainly as models show that they can differentiate the actions carried out in a basketball game with comparable success, with some actions having similar movement patterns, such as running and walking, mainly when considering the top-5 accuracy. The performance of I3D was also comparable, as the top-1 accuracy was a few percentage points below the published results and a few percentage points better on the top-5 accuracy (see Table 2). Pose-C3D struggled markedly on the dataset.

4.1 Analysis

The predictive analysis aims to find further insights into the model's performance. The confusion matrix shows the action classes that the model fared well with and the most problematic actions [20]. As shown in the confusion matrix for TSN on SpaceJam in Figure 2, the confusion matrix indicates a lack of separation in many instances. A noteworthy example is the 36 "ball in hand" instances that were predicted to be "dribble" as compared to the 34 that were correctly predicted, and a further 24 actions were predicted as "shoot" actions. Overall, the confusion matrix shows that the model did manage to correctly predict more times than negative for most of the actions.

Looking at the confusion matrix for I3D (Figure 3), the action class "no action" had the most success and performed even better than TSN. "Pass" scored the least number of correct predictions. It is also noteworthy that about half of the predictions fell into either "pick", "no action", or "walk".

Looking at the confusion matrix for PoseC3D (Figure 4) offers some insight into the performance. The distribution of predictions is concentrated across three actions, namely "pick", "no action" and "walk". The model managed to get only over 50 correct predictions. Many of the mispredictions fell into the "walk" category, which indicates that from a pose perspective, most of the samples resembled the walk action to the model. It is also noteworthy that after filtering the pose data, only 560 samples in the test set met the model "validity checks", which explains why the confusion matrix only has 560 predictions as opposed to 1270 in TSN and I3D above.

The remainder of the analysis will focus on the better-performing approach, TSN. To gain insight into the separation of the data, NCA is incorporated to enforce data clustering and make the data visually meaningful. Figure 5 shows the NCA representing the data clustering with the nearest neighbour set to 3, the random state set to 0 and the dimensionality set to 2.

To add to the confusion matrix of TSN, the precision, recall and f1-scores were also computed to gain more insight on a per-action level. Table 5 shows the calculated values. The number of

Actual \ Predicted	block	pass	run	dribble	shoot	ball in hand	defense	pick	no action	walk
block	70	0	6	2	38	0	4	7	0	0
pass	3	36	2	30	32	17	4	3	0	0
run	3	1	96	3	4	1	9	2	0	8
dribble	0	2	3	113	1	5	2	1	0	0
shoot	7	0	0	4	114	2	0	0	0	0
ball in han	6	12	8	36	24	34	4	3	0	0
defense	8	3	3	3	9	2	83	4	9	3
pick	5	1	2	2	6	1	9	97	1	3
no action	1	0	0	1	3	6	23	4	59	30
walk	9	4	10	2	7	2	9	13	12	59

Figure 2: Confusion matrix for TSN on SpaceJam Dataset.

Actual \ Predicted	block	pass	run	dribble	shoot	ball in hand	defense	pick	no action	walk
block	36	3	0	2	41	6	4	22	8	5
pass	2	11	6	16	17	25	1	28	13	8
run	2	2	34	31	7	1	3	19	7	21
dribble	0	2	21	55	5	2	2	20	12	8
shoot	4	0	0	3	92	11	0	8	4	5
ball in han	4	3	4	19	7	36	6	22	16	10
defense	3	0	0	15	1	9	23	24	39	13
pick	3	0	0	3	1	1	1	84	28	6
no action	0	0	0	0	1	3	1	4	110	8
walk	2	3	2	1	3	3	0	29	40	44

Figure 3: Confusion matrix for I3D on SpaceJam Dataset.

Actual \ Predicted	block	pass	run	dribble	shoot	ball in hand	defense	pick	no action	walk
block	0	0	0	0	0	0	0	13	5	37
pass	0	0	0	0	0	0	0	11	5	41
run	0	0	0	0	0	0	0	14	2	39
dribble	0	0	0	0	0	0	0	12	3	39
shoot	0	0	0	0	0	0	0	11	6	37
ball in han	0	0	0	0	0	0	0	8	10	45
defense	0	0	0	0	0	0	0	13	6	33
pick	0	0	0	0	0	0	0	8	3	44
no action	0	0	0	0	0	0	0	14	7	47
walk	0	0	0	0	0	0	0	7	3	37

Figure 4: Confusion matrix for PoseC3D on SpaceJam Dataset.

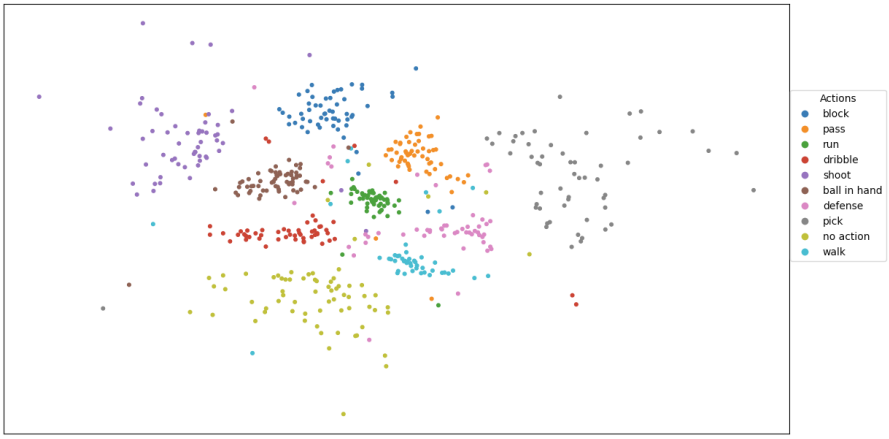


Figure 5: NCA for TSN with KNN = 3.

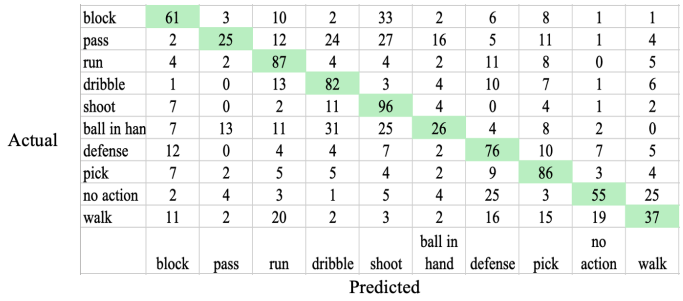


Figure 6: Confusion matrix for TSN ablation.

Table 5: Precision, Recall and F1-Score for TSN.

Action	Precision	Recall	F1-Score
block	0.55	0.63	0.59
pass	0.28	0.61	0.39
run	0.76	0.74	0.75
dribble	0.89	0.58	0.7
shoot	0.9	0.48	0.62
ball in hand	0.27	0.49	0.35
defence	0.65	0.56	0.61
pick	0.76	0.72	0.74
no action	0.46	0.73	0.57
walk	0.46	0.57	0.51

basketball actions is very restricted compared to the full scope of human actions, and even less so regarding SpaceJam, which has ten action classes. SpaceJam contains some actions that are closely related, such as running and walking. PoseC3D underperformed as pose approaches tend to need more training to get comparable results with the other models, and the 20 epochs for training were too restrictive and led to it being significantly outperformed by both TSN and I3D. The takeaway is the models' ability to learn/be pre-trained on a large action dataset such as Kinetics-400 and then further train the model in a specialised action recognition field such as basketball with a much-reduced dataset size is feasible; however, the make-up and quality of the data bares a large influence on the results.

4.2 Ablation

There is a considerable performance gap between TSN, I3D and Pose-C3D, so the ablation study will focus only on TSN. The ablation study will be conducted to see if there is an impact on the performance of a model should certain layers or components be removed. The objective is to see and understand the importance and purpose served by each element. Removing the frame resizing and cropping steps from the pipeline notably decreased as the top-1 and top-5 accuracies dipped to 41.26 and 84.96, respectively. Further changes to the pipeline steps and configuration resulted in a top-1 accuracy of 49.69 and 90.79 for the top-5 accuracy. In addition to removing the resizing and cropping steps, the batch size, clip length and frame interval were each reduced to 1, and data shuffling was enabled. Fig. 6 shows the confusion matrix for TSN after the changes. "dribble", "shoot", and "pick" suffer the biggest losses as the number of correct predictions all decreased by double digits.

5 Conclusion

The study shows the feasibility of an action recognition approach that works well within constrained data and limited compute resources environments (limited epochs). The research provides a foundation for future exploration of action recognition in basketball that can expand into play detection.

Future work could expand on the methods used in basketball footage. Furthermore, constructing a more robust dataset geared towards basketball with higher video quality can have a positive impact similar to what general action recognition datasets such as UCF-101 and HMDB-51 had.

The research highlighted some gaps in the field, such as the lack of a standard dataset that is truly focused on basketball actions, which is publicly accessible. The experiments provide insights into how the commonly applied action recognition approaches fare in the basketball context. TSN scored considerably higher than the other two and performed in line with previous results on other datasets. The SpaceJam dataset provides a step in the right direction, particularly when looking at the domain of basketball despite its shortcomings.

Looking at the confusion matrix for I3D (Figure 3), the action class "no action" had the most success and performed even better than TSN. "Pass" scored the least number of correct predictions. It is also noteworthy that about half of the predictions fell into either "pick", "no action", or "walk".

References

[1] Soomro, K., Zamir, A.R. (2014). Action recognition in realistic sports videos. In Computer Vision in Sports. 181–208. https://doi.org/10.1007/978-3-319-09396-3_9
[2] Özyer, T., A, D., Alhajj, R. (2021). Human action recognition approaches with video datasets - A survey. In Knowledge-Based Systems. 222, 106995. https://doi.org/10.1016/j.knosys.2021.106995
[3] Francia, S. (2015). SpaceJam: A Dataset for Basketball Action Recognition, https://github.com/simonefrancia/SpaceJam last accessed 05/06/2024.
[4] Chiura, T. B., van der Haar, D. (2023). Offensive Play Recognition of Basketball Video Footage Using ActionFormer. In Communications in Computer and Information Science, 1832, pp. 447-454. https://doi.org/10.1007/978-3-031-35989-7_57
[5] Onağ, Z., Tepeci, M. (2014). Team Effectiveness in sport teams: The effects of team cohesion, intra-team communication and team norms on Team Member satisfaction and Intent to remain. In Procedia - Social and Behavioral Sciences, 150, pp. 420–428. https://doi.org/10.1016/j.sbspro.2014.09.042

[6] Mallett, C.J., Lara-Bercial, S. (2016). Serial winning coaches: People, vision and environment. In Sport and Exercise Psychology Research, pp. 289 – 322. https://doi.org/10.1016/B978-0-12-803634-1.00014-5

[7] Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li, Z. (2017). A review on human activity recognition using vision-based method. In Journal of Healthcare Engineering vol. 2017, pp. 1–31. https://doi.org/10.1155/2017/3090343

[8] Host, K., Ivašić-Kos, M. (2022). An overview of Human Action Recognition in sports based on Computer Vision. In Heliyon, 8(6). https://www.cell.com/heliyon/fulltext/S2405-8440(22)00921-5

[9] Abdelkrim, N.B., Castagna, C., El Fazaa, S., El Ati, J. (2010). The effect of players' standard and tactical strategy on game demands in men's basketball. In The Journal of Strength & Conditioning Research, 24(10), pp. 2652-2662. https://doi.org/10.1519/JSC.0b013e3181e2e0a3

[10] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijaya-narasimhan, S. (2016). YouTube-8M: A Large-Scale Video Classification Benchmark, arXiv.org. https://doi.org/10.48550/arXiv.1609.08675

[11] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. (2014). Large-scale video classification with Convolutional Neural Networks. In Conference on Computer Vision and Pattern Recognition, pp. 1725 – 1732.

[12] Deng, J., Dong,W., Socher, R., Li, L., Li. K., Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In Conference on Computer Vision and Pattern Recognition, pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[13] Nisha, S. S., Sathik, M. M., Meeral, M. N. (2021). Application, algorithm, tools directly related to deep learning. Handbook of Deep Learning in Biomedical Engineering, Academic Press, pp. 61-84. https://doi.org/10.1016/B978-0-12-823014-5.00007-7

[14] Pan, Z.; Li, C. (2020). Robust basketball sports recognition by leveraging motion block estimation. In Signal Processing: Image Communication, 83, pp. 115784. https://doi.org/10.1016/j.image.2020.115784

[15] Liu, L. (2021). Objects detection toward complicated high remote basketball sports by leveraging deep CNN architecture. In Future Generation Computer Systems, 119, pp. 31 – 36. https://doi.org/10.1016/j.future.2021.01.020

[16] Nistala, A., Guttag, J. (2021). Using Deep Learning to Understand Patterns of Player Movement in the NBA. In MIT Sloan Sports Analytics Conference, pp. 1 – 14. https://api.semanticscholar.org/CorpusID:212413773

[17] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In Springer Cham, European conference on computer vision, pp. 20 – 36. https://doi.org/10.1007/978-3-319-46484-8_2

[18] Carreira, J., Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In Conference on Computer Vision and Pattern Recognition, pp. 6299-6308. https://doi.org/10.48550/arXiv.1705.07750

[19] Duan, H., Zhao, Y., Chen, K., Lin, D., Dai, B. (2022). Revisiting skeleton-based action recognition. In Conference on Computer Vision and Pattern Recognition, pp. 2969 – 2978. https://doi.org/10.48550/arXiv.2104.13586

[20] Heydarian, M., Doyle, T.E., Samavi, R. (2022). MLCM: Multi-label confusion matrix. In IEEE Access, 10, pp.19083-19095. https://doi.org/10.1109/ACCESS.2022.3151048