Video Activity Recognition Using Deep Learning

Sebastian Charmot
scharmot@stanford.edu

Mabel Jiang
jyh1998@stanford.edu

Rachel Yu
yuyue01@stanford.edu

Abstract

Quickly and accurately performing activity recognition on video input is a challenging problem with increasing demand and wide-spread applications. In this project, we leverage the Multi-Object Multi-Actor (MOMA) dataset to train, validate, and test on a two-stream model, where the first stream uses video classification, and the second stream uses object detection, and results from two streams is finally averaged. Strong performance of this two-stream model was demonstrated on video classification, and it is shown to outperform one-stream video classification and object detection model.

1 Introduction

We are interested in tackling the problem of recognizing activities from videos using only video as input. Unlike action recognition, which is characterized by simple motion patterns of a single person, activities are more sophisticated. Activities typically involve relationships and coordination between multiple agents, multiple objects, involve sequential steps, and a final goal [26]. As a result, developing models that can accurately and correctly predict activities is a challenging task.

The ability to automatically categorize human activities in videos is an area with increasingly important consequences from security and surveillance [18, 11] to entertainment [23], and health monitoring [4]. The rapid growth of video based social media apps such as TikTok, which has an estimated one billion active users, further emphasizes the importance of activity recognition from video. Recognizing activities from video in the domain of social media can improve recommender systems [1, 5], the relevance of advertisements [16], and also reduce exposure to harmful content [8, 21].

For our approach to video activity classification, we take the video as input and test multiple deep learning techniques to output an activity label for the input video. The next section describes the current main architectures and techniques for activity prediction from video. Section 3 describes the MOMA dataset and its features, which we use for our experiments. Section 4 describes the algorithms that we use to classify the activities in the MOMA dataset.

2 Background

Video recognition architectures can be separated into two main categories depending on whether the convolutional and layer operators use 2D (image-based) or 3D (video-based) kernels [2]. A visual of the outputs of 2D and 3D convolutions is illustrated below in Figure 1.

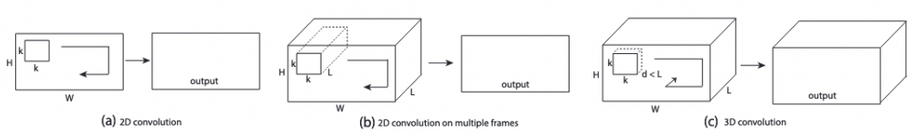


Figure 1: A Representation of 2D and 3D Convolutional Operations from [25]

2.1 Image Based Approaches

Using the fact that videos are simply a series of images, image based approaches attempt to extract features from individual frames and then combine these features into a singular prediction for the entire video [12]. These models are able to take advantage of the high performance of image classification networks and require significantly less resources to train than 3D based models due to a much smaller parameter space [2]. Although these models are well suited for identifying the objects in a video, one common problem that image based approaches face is that temporal information is not captured [14, 17]. As a result, the sequence of events is not taken into consideration, which can potentially be used to distinguish one activity from another. To account for temporal information and change, some image based researchers include a recurrent layer, such as an LSTM layer, to capture temporal ordering and long range dependencies [6]. This allows the model to learn sequential dynamics and make better predictions on videos with sequential structure. Image based approaches are also susceptible to camera motion. Researchers have highlighted the negative effect of camera motion on image based approaches, and adjustments for camera motion resulted in better predictions [27].

2.2 Video Based Approaches

Video based approaches leverage 3D convolutional layers, which contain spatio-temporal filters allowing them to directly create hierarchical representations of spatio-temporal data [2]. 3D convolutional neural networks (CNNs) have been shown to be more suitable for spatio-temporal feature learning compared to 2D ConvNets [25]. By using 3D CNNs, researchers were able to achieve marginally better results than both traditional 2D approaches and 2D approaches with recurrent layers [25] on the UCF101 dataset [24]. More

recent approaches include SlowFast, which is a type of architecture for video recognition models introduced by researchers at Facebook AI in 2019 [7]. Its name comes from the fact that it utilizes two streams of 3D CNNs that operate at different temporal resolutions. The "slow" stream processes video frames at a lower frame rate, using a larger receptive field to capture long-term temporal information. The "fast" stream processes video frames at a higher frame rate, using a smaller receptive field to capture short-term temporal information. The outputs of the two streams are then combined and fed into a classification layer, which outputs the activity prediction. By leveraging these two streams to incorporate temporal information from video, SlowFast reported state-of-the-art accuracy on major video recognition benchmark datasets, Kinetics [13], Charades [22], and AVA [9].

3 Dataset

For our project, we leverage the Multi-Object Multi-Actor Activity (MOMA) data-set [15], which contains 20 activity categories, 91 sub-activity categories, 226 object categories, 26 actor categories, and 52 relationship (static and dynamic) categories. It is the first video-based dataset with multi-object, multi-actor, and categorical labels for actors (i.e., social roles) and objects, which provide exhaustive details for the associated activity. This information is spread across 373 raw videos at the activity level with a combined play time of 148 hours. At the frame level, the dataset contains hypergraph annotations for 37,428 frames, with 164,162 actor/object instances of 20 actor classes and 120 object classes. On average, there are 4.39 actors/objects and 3.18 higher-order relationships per frame, 5 instances of atomic actions per clip, and 6.34 instances of sub-activities per untrimmed video.

The reason that we chose to use the MOMA dataset is that it is the video dataset with the most information about object and actor categories. Almost all video datasets do not have the detailed frame-wise annotated data that MOMA contains. In Figure 2, we illustrate the information contained within a frame of the MOMA dataset. We want to leverage this frame-wise data in various single and two-stream networks to investigate the effects of including information about the objects and actors found in videos as well as spatio-temporal information on activity prediction. The dataset is split into training set of size 1130, validation set of size 282, and test set of size 282.

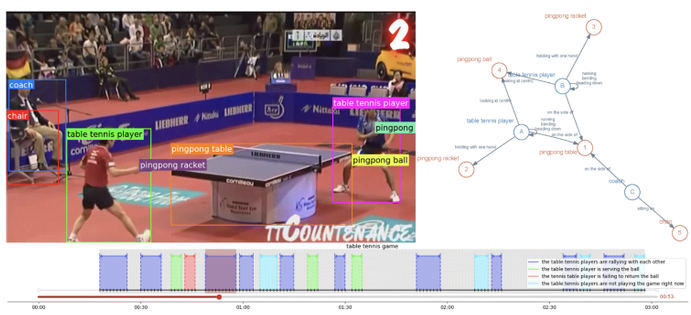


Figure 2: Example Frame Selected from the MOMA Dataset [15]

4 Methods

We propose a two-stream human activity classification model that leverages both spatio-temporal information and actor, object information. Our first stream consists of 3D CNN video classification models to capture spatio-temporal information directly from video. Our second stream consists of 2D based object detection models that will leverage information about the objects and actors within the frames of each video. We merge these two streams to make a final prediction. This architecture is illustrated in Figure3. Our

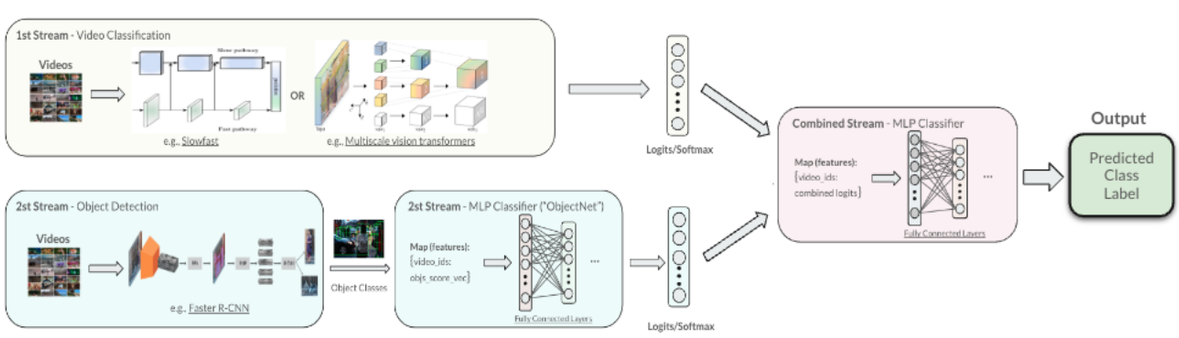


Figure 3: Illustration of Our Proposed Two-Stream Architecture

method is inspired by video activity classification models in recent literature that tend to have a two-stream architecture, where each stream provides its own benefits [2]. Because the MOMA dataset contains such detailed object and actor annotations, we have the unique ability to test a 2D based object detection model as one of our streams.

For our two-streamed model, we design the Softmax function as the activation function in the output layer, use cross-entropy loss as shown in Equation (1) and SGD optimizer to train the model

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{1}$$

where M is the number of classes, y is binary indicator if class label c is the correct classification for observation o, p is predicted probability observation o is of class c

5 Experiments

Architecture	ResNet-18	SlowFast	MViT
Backbone Input	Image/Frame	Video	Video
Feature Extraction	method 1: get middle frame method 2: train: random frame val and test: middle frame	train: randomly sampled clips val and test: constant sampled clips	train: randomly sampled clips val and test: constant sampled clips
Transformation	cropping, flipping, and normalization	cropping, flipping	cropping, flipping

Tabel 1. Implementation Details of Architectures in First Stream

Model	GraphNet	Object Detection	"ObjectNet"	Last MLP Classifer
Backbone Input	Graph	Image	Object	Combined logits
Feature Extraction	actors, objects relationships, and attributes	individual objects and actors in a frame	actors and objects in a video	tensors (concatenated logits)

Tabel 2. Implementation Details of Architectures in Second Stream

5.1 First Stream/Baselines: Video Classification

For our baselines, we have two different approaches to classify a video, image-based approach, using ResNet-18, and video-based approach, using SlowFast, and Multiscale Vision Transformers (MViT).

ResNet-18 is a convolutional neural network that has 18 deep layers, which takes an image as an input and classifies the image to one of the 1000 class labels. As seen in Table 1, there're two different feature extraction methods for ResNet-18, one is extracting the middle frame in video for entire dataset, and the other method is to get the middle frame only for test dataset while get a random frame for training set. The image is then transformed into tensor using the described method in Table 1 and fed into our model.

Using SlowFast, MViT as our video-based approach backbones, both of them take videos as input and output a video class label. Different approaches to select clips of the video and how we transform sampled clips can be found in Table 1.

5.2 One Exploration: "GraphNet"

Graph features in the MOMA data-set include actors, objects, relationships, and attributes in a video that can be extracted. The features are then concatenated and fed into a Multi-layer Perceptron classifier (we named it "GraphNet"). This step of exploration using "GraphNet" helps us identify what information

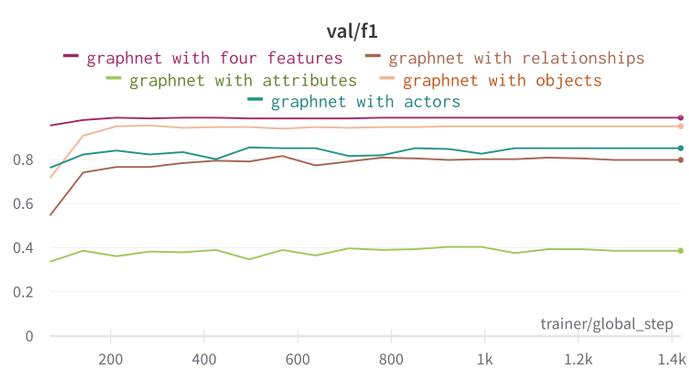


Figure 4: F1-Score of "Graphnet" with Different Input

among those extracted features can be the most helpful in classifying a video and therefore prepares us for the second stream in our model. That explains why ground truth is used in training "GraphNet". Figure 4 shows the F1-score of "GraphNet" using ground truth labels, trained on four features separately, and all four features. It can be seen that "GraphNet" has the best performance when it's trained on all four features,

second to best on only object features, which obtain just slightly lower accuracy than the one with "all four features", and third to best on only actor features. Therefore, we can say that objects and actors are the relatively most helpful information in predicting class labels.

From the insights above, we decided to exploit object detection as an intermediate step in our non-end-to-end model. We expect that extra information about the objects in a video will improve the performance.

5.3 Second Stream: Object Detection

To perform the object detection stream, we implemented the Faster R-CNN architecture [20] using MMDectection [3]. The input to this model is an image, and it outputs a series of actors, objects, their bounding boxes, and corresponding confidence values that it identifies in that image.

Figure 5: Example Frame Selected from the MOMA Dataset [15]

5.4 Second Stream: MLP Classifier - "ObjectNet"

We aggregate the prediction results from object detection across several selected frames by summing the confidence values of each object. Then, our MLP classifier, which we refer to as "ObjectNet", takes in this object-wise aggregated confidence as input, and feeds it into two fully connected layers to output a video label.

5.5 Combined Two-Stream Model

After we implemented two streams, we collect the logits or softmax vectors from two streams, then we purpose two methods to implement the merging step to get the final result.

- Simply combination: taking the average of two logits/softmax and getting the prediction by taking argmax function.
- Build another MLP classifier: takes as input the concatenation of logits/softmax returned from both streams, and outputs the predicted video class label.

5.6 Hyperparameters and Metrics

For learning rate, we used a cosine annealing learning rate schedule with an initial learning rate of 0.0005, so that we start with a large learning rate that is then relatively rapidly decreased to a minimum value before being increased rapidly again. Empirically cosine learning rate has performed well on a number of tasks[19], like image classification, since a fluctuating learning rate up and down can prevent us from getting stuck in a sub-optimal area, given that a neural network is always non-convex. Batch size is set to 8 and number of epochs is 30, where we followed the setting of batch size and numerb of epochs specified in SlowFast[7] architecture.

Accuracies for top 1 prediction and top 5 predictions and F1-Score are the metrics in evaluating the performance of our model, given that they're the most commonly used metrics in the classification model. Accuracy estimates the percentage of all correctly classified observations, while F1-score calculates the harmonic mean of precision and recall.

6 Results and Discussion

6.1 Results Analysis

Baseline results can be found in Table 3, where we can conclude that all models perform better when they're pretrained. For ResNet-18, it performs the best when the middle frame from a video is selected for the training set, rather than the random frame. SlowFast and MViT performed better than the image-based model in the aspect of classification accuracy and F1-Score. The highest performance is from MViT with

Model	Pretrain	Metrics			Frame Selection
		acc1	acc5	F1	
ResNet-18 [10]	None	0.3239	0.6479	0.2007	Middle
	ImageNet	0.5775	0.8310	0.4632	Random
	ImageNet	0.5915	0.8169	0.4248	Middle
SlowFast r50 [7]	None	0.4397	0.7411	0.3005	N/A
	Kinetics-400	0.7730	0.9468	0.6379	N/A
MViT	None	0.3017	0.5907	0.1507	N/A
	Kinetics-400	0.8509	0.9740	0.7563	N/A

Table 3. Baselines Results

pre-train weight, which is 0.85 for accuracy@1 and 0.75 for F1 score. So two video-based models are selected to implement our method.

Our method result can be found in Table 4, where we can conclude that by simply merging which is the method 1, the results are higher than any one of a single stream. The method 2 highlighted in pink, which are combining logits with a simple MLP classifier and shows in the last two rows of the table, they have the highest result for all metrics. Among them, the model with MViT backbone has the best performance.

Method	Model	Metrics		
		acc1	acc5	F1
Single stream	SlowFast w/ Pre-train	0.7730	0.9468	0.6379
	MViT w/ Pre-train	0.8509	0.9740	0.7563
	"ObjectNet"	0.8121	0.9645	0.7088
Simply merge	SlowFast + "ObjectNet"	0.8298	0.9574	0.7192
	MViT+ "ObjectNet"	0.8759	0.9752	0.7964
Combined +MLP classifer	SlowFast + "ObjectNet" + MLP	0.8617	0.9787	0.7637
	MViT+ "ObjectNet" + MLP	0.8961	0.9823	0.7986

Table 4. Our Method Results

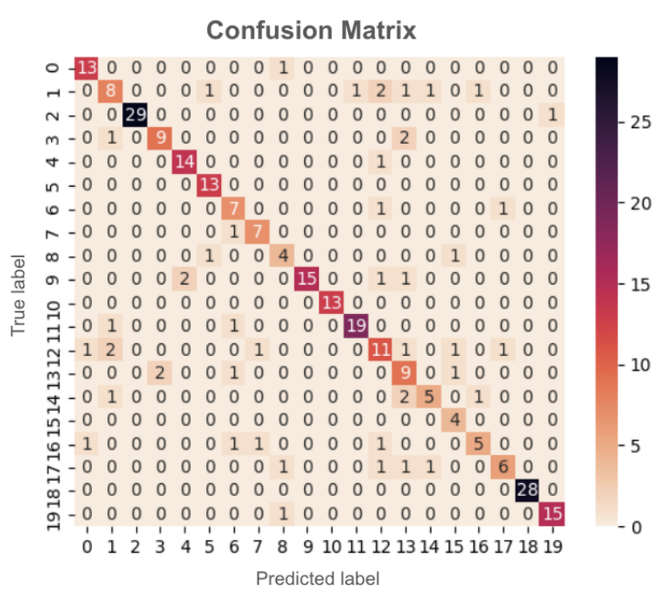


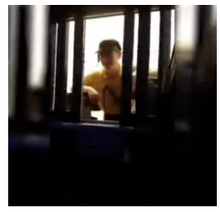
Figure 6: Confusion Matrix of the Best Result

6.2 Error Analysis

Three failed predictions of videos were selected for error analysis. The true labels, predicted labels, and screenshots of the videos are shown in Figure 7. The common issue for these videos is delivering ambiguous information, which is even hard for humans to conclude specific labels to them in real life. To be specific, there are too many people walking around in hospitals in video 1 (medical injection) and it contains many scenes such as doors, rubber gloves, and getting in lines, which are similar to scenes of security screening - its predicted label. In addition, the confusion matrix (shown in Figure 5) also shows that videos labeled as "medical injection" are highly likely to be predicted wrongly. In video 2, the resolution and brightness are too low to see the whole scene clearly, while the server inside of the store is under more light so that section of the scene is easier to be captured and emphasized by the model and predicted as a reception service. As

for video 3, the background is too messy to figure out where the actors are. From the above analysis, we infer that our model is likely to fail on videos with too complicated scenes, low resolution, dark environments, and messy backgrounds.

(a) Screenshot from Video 1
True Label: medical injection
Predicted Label: security screening



(b) Screenshot from Video 2
True Label: drive-thru ordering
Predicted Label: reception service

(c) Screenshot from Video 3
True Label: dining
Predicted Label: haircut

Figure 7: Examples of failed predictions

7 Conclusion and Future Work

Our results indicate that the two-stream model with the last MLP classifier outperforms the other models including one-stream models, achieving an accuracy of 0.89 and an F1-score of 0.79. This suggests that the two-stream architecture provides more accurate predictions for video classification. Therefore, we can conclude that combining the result from video classification and object detection gives a decent prediction of the video.
In the future, if given more time, instead of object detection, the second stream can be explored with graph convolution outputting graph information, which contains more information than object detection. Finally, a dynamic scene graph generation could be done to offer a dynamic representation of the actors and objects in a video, as well as their attributes and relationships.

8 Contributions

All baselines and simply merging implementations: Rachel Yu, and one of my teammate in CS229
MViT baselines and combined MLP classifier model training: Rachel Object detections: the last teammate in CS229

References

[1] Anitha Anandhan et al. "Social Media Recommender Systems: Review and Open Research Issues". In: *IEEE Access* 6 (2018), pp. 15608–15628. DOI: 10.1109/access.2018.2810062. URL: https://doi.org/10.1109/access.2018.2810062.

[2] Joao Carreira and Andrew Zisserman. *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*. 2017. DOI: 10.48550/ARXIV.1705.07750. URL: https://arxiv.org/abs/1705.07750.

[3] Kai Chen et al. *MMDetection: Open MMLab Detection Toolbox and Benchmark*. 2019. DOI: 10.48550/ARXIV.1906.07155. URL: https://arxiv.org/abs/1906.07155.

[4] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. "Transfer learning for activity recognition: a survey". In: *Knowledge and Information Systems* 36.3 (June 2013), pp. 537–556. DOI: 10.1007/s10115-013-0665-3. URL: https://doi.org/10.1007/s10115-013-0665-3.

[5] Yashar Deldjoo et al. "Recommender Systems Leveraging Multimedia Content". In: *ACM Computing Surveys* 53.5 (Oct. 2020), pp. 1–38. DOI: 10.1145/3407190. URL: https://doi.org/10.1145/3407190.

[6] Jeff Donahue et al. *Long-term Recurrent Convolutional Networks for Visual Recognition and Description*. 2014. DOI: 10.48550/ARXIV.1411.4389. URL: https://arxiv.org/abs/1411.4389.

[7] Christoph Feichtenhofer et al. "SlowFast Networks for Video Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[8] Vaishali U. Gongane, Mousami V. Munot, and Alwin D. Anuse. "Detection and moderation of detrimental content on social media platforms: current status and future directions". In: *Social Network Analysis and Mining* 12.1 (Sept. 2022). DOI: 10.1007/s13278-022-00951-3. URL: https://doi.org/10.1007/s13278-022-00951-3.

[9] Chunhui Gu et al. *AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions*. 2017. DOI: 10.48550/ARXIV.1705.08421. URL: https://arxiv.org/abs/1705.08421.

[10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[11] Cheng-Bin Jin, Shengzhe Li, and Hakil Kim. *Real-Time Action Detection in Video Surveillance using Sub-Action Descriptor with Multi-CNN*. 2017. DOI: 10.48550/ARXIV.1710.03383. URL: https://arxiv.org/abs/1710.03383.

[12] Andrej Karpathy et al. "Large-Scale Video Classification with Convolutional Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2014. DOI: 10.1109/cvpr.2014.223. URL: https://doi.org/10.1109/cvpr.2014.223.

[13] Will Kay et al. *The Kinetics Human Action Video Dataset*. 2017. DOI: 10.48550/ARXIV.1705.06950. URL: https://arxiv.org/abs/1705.06950.

[14] Ivan Laptev et al. "Learning realistic human actions from movies". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2008. DOI: 10.1109/cvpr.2008.4587756. URL: https://doi.org/10.1109/cvpr.2008.4587756.

[15] Zelun Luo et al. "MOMA: Multi-Object Multi-Actor Activity Parsing". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 17939–17955. URL: https://proceedings.neurips.cc/paper/2021/file/95688ba636a4720a85b3634acfec8cdd-Paper.pdf.

[16] Tao Mei et al. "VideoSense: Towards Effective Online Video Advertising". In: *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*. ACM Press, 2007. DOI: 10.1145/1291233.1291467. URL: https://doi.org/10.1145/1291233.1291467.

[17] Juan Carlos Niebles, Hongcheng Wang, and Fei-Fei Li. "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words." In: vol. 79. Sept. 2006, pp. 1249–1258.

[18] Sangmin Oh et al. "A large-scale benchmark dataset for event recognition in surveillance video". In: *CVPR 2011*. IEEE, June 2011. DOI: 10.1109/cvpr.2011.5995586. URL: https://doi.org/10.1109/cvpr.2011.5995586.

[19] E. Raff. *Inside Deep Learning: Math, Algorithms, Models*. Manning, 2022. ISBN: 9781617298639. URL: https://books.google.com/books?id=s8hhzgEACAAJ.

[20] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. DOI: 10.48550/ARXIV.1506.01497. URL: https://arxiv.org/abs/1506.01497.

[21] Morgan Klaus Scheuerman et al. "A Framework of Severity for Harmful Content Online". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW2 (Oct. 2021), pp. 1–33. DOI: 10.1145/3479512. URL: https://doi.org/10.1145/3479512.

[22] Gunnar A. Sigurdsson et al. *Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding*. 2016. DOI: 10.48550/ARXIV.1604.01753. URL: https://arxiv.org/abs/1604.01753.

[23] Khurram Soomro and Amir R. Zamir. "Action Recognition in Realistic Sports Videos". In: *Computer Vision in Sports*. Springer International Publishing, 2014, pp. 181–208. DOI: 10.1007/978-3-319-09396-3_9. URL: https://doi.org/10.1007/978-3-319-09396-3_9.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. *UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild*. 2012. DOI: 10.48550/ARXIV.1212.0402. URL: https://arxiv.org/abs/1212.0402.

[25] Du Tran et al. *Learning Spatiotemporal Features with 3D Convolutional Networks*. 2014. DOI: 10.48550/ARXIV.1412.0767. URL: https://arxiv.org/abs/1412.0767.

[26] Pavan Turaga et al. "Machine Recognition of Human Activities: A Survey". In: *IEEE Transactions on Circuits and Systems for Video Technology* 18.11 (Nov. 2008), pp. 1473–1488. DOI: 10.1109/tcsvt.2008.2005594. URL: https://doi.org/10.1109/tcsvt.2008.2005594.

[27] Heng Wang and Cordelia Schmid. "Action Recognition with Improved Trajectories". In: *2013 IEEE International Conference on Computer Vision*. IEEE, Dec. 2013. DOI: 10.1109/iccv.2013.441. URL: https://doi.org/10.1109/iccv.2013.441.