IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 October 2024, accepted 20 October 2024, date of publication 23 October 2024, date of current version 1 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3485199

RESEARCH ARTICLE

Synthetic Data Augmentation for Video Action Classification Using Unity

NINO CAULI AND DIEGO REFORGIATO RECUPERO
Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy
Corresponding author: Diego Reforgiato Recupero (diego.reforgiato@unica.it)

This work was supported by European Union's Horizon 2020 Marie Skłodowska-Curie Actions Individual Fellowships under Grant 101031646.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of the University of Cagliari.

ABSTRACT In video analysis, collection and labeling of data can be time and resource-consuming. To solve the scarcity of data problems, synthetic data augmentation is a promising solution. In this paper, we present an approach to generate synthetic videos for action recognition using Unity, the popular game engine. The synthetic videos are generated with high variability in lighting, subjects' models, backgrounds, animations, and camera positions. We use the generated data to augment a small dataset of subjects who are executing physical exercises for action recognition. We tested the augmented data on two state-of-the-art models for action classification and demonstrated the significant benefits of synthetic data augmentation for improving the performance of these models on small datasets in the context of video action recognition.

INDEX TERMS Data augmentation, action recognition, convolutional neural networks, video transformers, synthetic video generation.

I. INTRODUCTION

In the past 10 years, we have entered the era of Artificial Intelligence (AI). The increasing amount of data available as a result of the Internet and social media, coupled with the exponential increase in computational power, has given rise to AI models capable of solving complex tasks and assisting individuals in their daily activities. The most common type of data available is written text, images, and videos. In particular, images, and to a greater extent, videos, contain a wealth of intrinsic information that can be harnessed by well-crafted and trained AI models. The high availability of images and videos can be attributed to the prevalence of cameras in our surroundings, including smartphones, webcams, surveillance systems, car sensors, drones, robots, etc.

Classical Image Processing (IP) and Computer Vision (CV) models are quickly being replaced by Convolutional Neural Networks (CNN), Video Transformers (ViT), or other Deep Learning (DL) models [1], [2], [3]. The extensive set of trainable parameters of DL models requires a massive amount of data to learn a task. While written text is easy to collect and store, the acquisition and storage of images and videos are trickier. Collecting video images from real cameras can be very time-consuming, and the storage space needed for image data tends to be massive. Often, video data are protected by strict privacy policies, especially when recorded in public areas with human subjects involved. Medical data is the clearest example of protected data, and it can be tricky or even impossible to obtain past recordings from hospitals. In areas such as robotics and autonomous vehicle control, long video collection sessions result in wearing or damaging mechanical components and dangerous interactions between machines and operators. Last but not least, labeling these huge image/video datasets is a troublesome process. When automated labeling is not an available option, all the data needs to be manually labeled by humans. This process is very time-consuming and, in some cases, particularly tricky (e.g., object 3D pose estimation, image segmentation, frame-by-frame video labeling, etc.). For these reasons, datasets for

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

specific IP and CV problems are often small and, in the case of classification datasets, unbalanced. Fortunately, for more generic and common CV tasks (e.g., object recognition, face recognition, autonomous driving, etc.), vast datasets already exist [4], [5], [6], [7]. DL models are often pretrained on those big generic datasets and then fine-tuned on small datasets collected for a specific IP or CV task [8].

Pretraining helps in situations of data shortage, but it does not address the problem directly. With data augmentation, on the other hand, new data are generated artificially from a similar distribution as the original dataset to be augmented. Classical data augmentation techniques for images include cropping, flipping, rotating, translating, and alterations to histograms and RGB values [9], [10]. More recent approaches use Generative Adversarial Networks (GANs) to generate new augmented images [11], [12], [13], [14], [15]. After training the GAN on an unlabelled dataset, the generator can be used to augment the original dataset with newly generated images from the same statistical distribution as the original dataset.

These data augmentation approaches use images from the original dataset as a base for generating the new augmented images. A different strategy is to generate the images for the augmented dataset from a simulation of the physical world (e.g., environment, world physics, and cameras). The simulated scenes are easy to randomize, and a dataset of synthetic images can be generated [16]. RGB cameras generate high-level data filled with information. For this reason, simulating camera sensors is complex, and detailed graphical and physical simulators are needed to generate realistic images. Modern game engines, such as Unity [17] and Unreal Engine [18], among others, are good simulator candidates, capable of rendering photorealistic images in just a few milliseconds, simulating realistic physical interactions between objects, and offering powerful scripting and design tools to recreate detailed artificial scenes [16], [19].

Although some researchers have already developed image data generators based on game engines for object recognition and detection [20], there are few solutions for generating synthetic videos specifically tailored for action recognition, and these are often designed for highly specific scenarios [21]. In this paper, we introduce a synthetic video generator created in Unity capable of producing videos featuring a synthetic avatar performing preselected actions in front of a simulated camera. The generated videos possess various degrees of randomization, including background image, lighting intensity and color, camera position, animation speed, and avatar appearance. Furthermore, the videos are automatically labeled with action classes and skeletal data. We tested the generator for recognizing gentle gymnastic exercises performed by a human subject in videos. Initially, we recorded a small dataset of real subjects executing a set of gentle physical exercises and then augmented it with a dataset of synthetic videos generated using our synthetic video generator. Both the real and augmented datasets were used to train two state-of-the-art video action recognition models (I3D [22] and Timesformer [23]).

This work is part of a European project called DrVCoach, which aims to develop a robotic coach for monitoring and motivating seniors in performing a daily gentle physical exercise routine to maintain their physical fitness as they age.[1]

The remainder of the paper is organized as follows: Section II presents existing related works on video action recognition and data augmentation; Section III describes in detail the proposed synthetic data generator; Section IV defines the task on which we tested the generator; Section V presents the results obtained on I3D and Timesformer models; and Section VI draws the conclusions.

II. RELATED WORKS

In image action recognition, two main sub-tasks must be addressed: action representation and action classification [24]. Action representation involves extracting significant features from images for action classification, while action classification is the process of categorizing actions based on these features. Early action recognition algorithms tackled these tasks separately, utilizing handcrafted features to represent actions [25], [26], and employing standard approaches, such as Support Vector Machines (SVM) and k-means, for recognition. However, in recent years, with advancements in computational power and the availability of larger video action recognition datasets [27], [28], [29], DL models have emerged as a comprehensive solution that combines action representation and classification. Over the past decade, CNNs have become the de facto standard for image recognition [30], [31], [32]. Nevertheless, in the last three years, models based on ViT architecture [33] are quickly replacing CNNs [23], [34], [35], [36]. For a comprehensive review of video transformers, please refer to the work of Selva et al. [3].

When transitioning from images to videos, the time dimension is added to space and color dimensions. Videos are sequences of images that unfold over time. While models designed for image analysis can be used for video analysis, it is essential to consider the temporal dimension inherent in videos. For this reason, state-of-the-art models for video analysis are designed to capture temporal information. These models include Optical flow based methods [31], 3D CNNs [30], Recurrent Neural Networks (RNN) [32], [37] and ViT with space-time attention [23].

As we have already pointed out, CNNs and ViT are the most popular models in image analysis. These models require a substantial amount of data for successful training. Several data augmentation techniques for image analysis have been introduced and well-presented in recent review papers by Shorten and Khoshgoftaar [9], Khalifa et al. [10]. These surveys provide a comprehensive overview of image data augmentation, covering basic image manipulations such as

[1] https://drvcoach.unica.it/

geometrical and color space transformations, kernel filters, noise injection, mixing images, and random erasing. They also delve into more recent DL approaches, including feature space augmentation, adversarial training, GAN-based techniques, and Neural Style Transfer.

A different approach involves the generation of synthetic images using simulators capable of emulating the appearance and physics of the real world. As an example, authors in [38] developed and evaluated synthetic human data for improving human action recognition, particularly with regard to generalization across unseen viewpoints. The new data generation methodology they proposed is called SURREACT and allows the training of spatio-temporal CNNs for action classification. The synthetic videos are created by superimposing one or more virtual avatars performing the desired actions onto a background that represents a real scene. While this approach is similar to ours, we believe it is less flexible. By using a game engine to create the virtual scene, we can easily modify lighting parameters and, in future work, update our model by adding objects to the scene (e.g., chairs, tables) and incorporating avatar-object interactions. Another example is provided by [39], where the authors explored the generation of synthetic training data for action recognition. Specifically, they introduced an interpretable parametric generative model for human action videos, leveraging procedural generation techniques and advanced computer graphics capabilities from modern game engines. By combining their extensive set of synthetic videos with small real-world datasets, they achieved high recognition performance. The actions used to generate the synthetic data were procedurally created by combining Motion Capture (MoCap) data with programmatically defined actions. Although MoCap data is highly accurate, MoCap setups are costly and time-consuming to implement. In our approach, we extract 3D action parameters directly from 2D videos, eliminating the additional time and costs associated with the motion capture process.

Game engines are often employed for generating new images due to their powerful graphic and physics engines. Unreal Engine, for instance, was used by Tremblay et al. [16] to create a photorealistic dataset for object detection. The authors were able to vary several features of the images, such as object positions, background, illumination, and camera positions. The concept of producing a synthetic dataset with high variability in the scene depicted in the generated images is emphasized by the Domain Randomization (DR) paradigm introduced by Tobin et al. [19]. In DR, the simulated scene parameters need to be highly randomized to generate images that fully cover the data distribution to be learned. Augmented datasets generated in this way contain a high variability in lighting, object shapes, textures, camera positions, and physics behaviors. When dealing with videos, it is essential to consider the correlation between time and space, not only in the design of learning models but also in the generation of training datasets. For instance, geometric and color space transformations must be kept consistent throughout an entire video sequence. In GAN-based and Neural Style Transfer data augmentation techniques, generators can be implemented using time-domain-specific models, such as 3D CNNs or convolutional RNNs. In simulations, on the other hand, the physical interaction between objects (e.g., rigid bodies interaction, and gravity), their motions, and the animation of subjects in the scene become crucial aspects for the generation of synthetic videos [40].

While several large RGB video datasets for generic action recognition tasks already exist [41], [42], [43], [44], [45], [46], [47], [48], [49], for more specific action recognition tasks, additional data may need to be collected. One solution is fine-tuning the DL model on these large generic video datasets. A different approach to address the shortage of data is the use of data augmentation techniques specific to videos. The recent survey titled ''Survey on videos data augmentation for deep learning models'' [40] provides a deep analysis of video data augmentation through simulation.

In contrast with the mentioned approaches, in this paper, we introduce a new synthetic video generator specifically designed for action recognition. Our decision to develop this tool arose from the absence of readily available and user-friendly solutions capable of meeting our unique randomization requirements. While some existing solutions address object detection in static images, such as the Unreal Engine 4 plugin 'NVIDIA Deep learning Dataset Synthesizer (NDDS)' [20], and others offer tools for generating synthetic videos for action recognition, such as ElderSim, a platform for synthetic data generation focusing on human action recognition within house interiors [21], none of these options provides the level of flexibility and randomization required for our augmented action recognition dataset.

III. SYNTHETIC VIDEO GENERATOR

In this paper, our primary objective is to address the challenge of action recognition from videos. To overcome the scarcity of data for this task, we have developed a video generator capable of producing synthetic videos featuring a subject performing specific actions. Indeed, our tool allows us to add new skeletal animations, randomize background skyboxes using user-selected HDRI images, procedurally randomize avatar appearance and age, and introduce randomization in lighting, camera position, and animation speed. This versatile tool serves a dual purpose: it can augment preexisting datasets, enriching them with diverse examples, or create entirely new datasets tailored to specific research needs. Our generator is implemented using the Unity game engine [17] along with the Perception [50] and Synthetic Humans [51] packages.

Game engines are optimal tools to generate personalized scenarios, simulate real world physics, and render realistic images. Unity, together with Unreal Engine, is one of the most used game engine for game development, design and research. Unity in particular stands out from the competition for several reasons such a user friendly interface, a complete and practical C# scripting API, cross platform building

FIGURE 1. Example of a frame rendered by the synthetic video generator.

options, a large online asset store, and a free to use plan for projects without revenue. Moreover, Unity posses several plugins specifically develop for researchers in Computer Vision, Artificial Intelligence and Robotics. Unity Perception Package is a toolkit for generating synthetic datasets for computer vision. It offers a set of predefined and customizable scene randomizers, automatic labelling tools, and a C# library to customize all the parameter of the data generator. Synthetic Humans, on the other hand, is a plugin to procedurally generate and place human avatars in a virtual scene.

The videos generated by our tool feature a virtual humanoid avatar placed in a highly randomized scene, with the avatar performing one action randomly selected from a pool of preselected actions. The pool of actions used in our tests was derived from real videos using the pose landmark detection pretrained model from the MediaPipe library [52].

MediaPipe Solutions is a suite of libraries, models, and tools for applying AI and ML to vision, text, and audio analysis. MediaPipe's pose landmark detection model is capable of detecting body pose landmarks in image coordinates and in 3D world coordinates from images or videos. We created a few scripts to extract the 3D body pose from videos of a human subject performing gentle gymnastic exercises, which we later used to generate Unity animation clips for our generator.

Figure 1 illustrates a frame rendered by our synthetic video generator. The scene features a human avatar engaged in an exercise, set against various indoor or outdoor environments. The background is created using a skybox with a randomly selected HDRI texture. The textures used in this study are selected from the 'HDRI Haven' Indoor, Nature, and Urban free assets available in the Unity Asset Store.[2]

The HDRI background plays a crucial role in the scene's global illumination, complemented by the inclusion of a directional light with adjustable color and intensity. The virtual camera responsible for capturing frames consistently faces the avatar. However, we introduce randomization in various aspects, including its distance from the avatar, vertical and horizontal translations, longitudinal axis rotation, and 3D spatial positioning.

Additionally, the avatar itself is procedurally generated using the Synthetic Humans Unity package. This package has the capability to create human avatars with randomized features, such as age, gender, ethnicity, height, weight, and clothing, all derived from the available asset pools. For this study, we generate avatars using the default asset pool provided by the package. For each video generated, the synthetic video generator provides the following outputs:

1) A.png image for each frame of the video. The resolution of the images is an adjustable parameter of the generator.

[2]https://assetstore.unity.com/publishers/49283

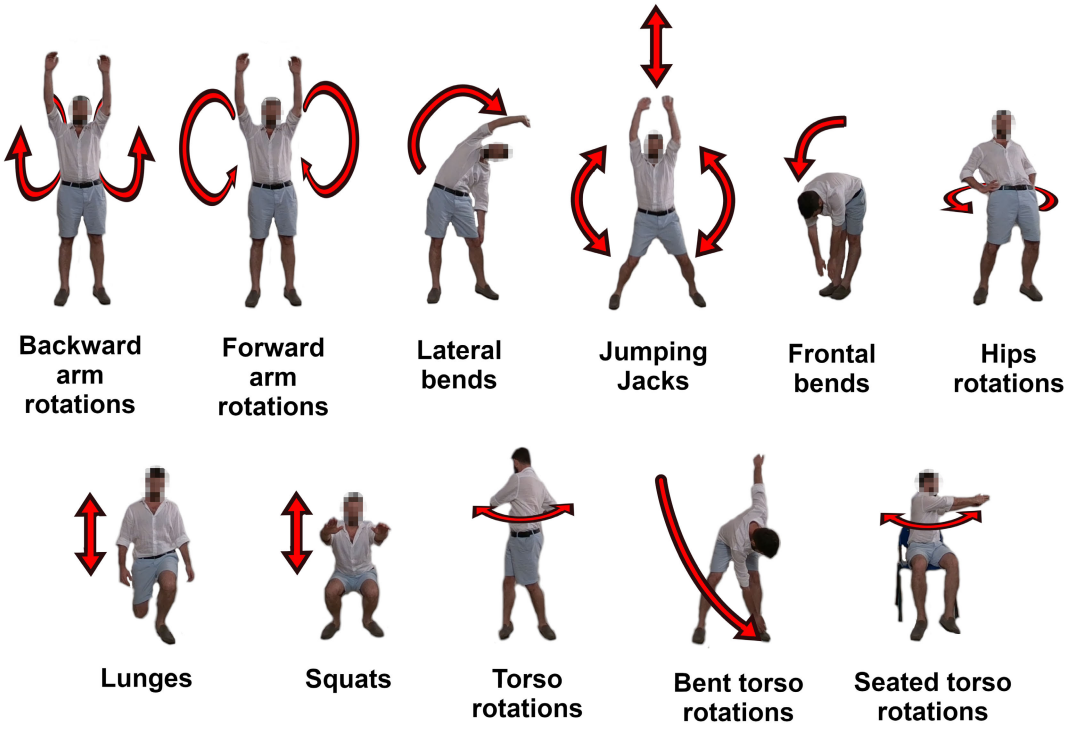


FIGURE 2. The 11 exercises used as classes for our datasets.

2) A.csv file for each frame that includes information on various labels automatically calculated by the generator. This information encompasses the name of the action, 3D camera pose, 3D global pose of each joint of the avatar, 2D pose of each joint in camera space, and avatar metadata (age, ethnicity, etc.).

The synthetic video generator and the MediaPipe scripts for extracting skeletal animations from videos and converting them into Unity animation clips are accessible online through the provided link in the footnote.[3]

IV. VIDEO ACTION RECOGNITION TEST

To evaluate the performance of our Synthetic Data Generator, we conducted an action recognition task. The objective of this task was to recognize one out of the 11 gentle gymnastic exercises performed by a human subject in a video. Figure 2 displays the 11 exercises classified during the test, selected in consultation with a professional personal trainer. Gentle gymnastics represents a form of physical activity that involves slow and progressive movements designed to mobilize the entire body. These exercises are suitable for individuals with varying levels of training, including sedentary office workers, older adults, and athletes. Regular participation in gentle gymnastics offers several advantages, including weight management, posture enhancement, and muscle toning. For older adults, a daily routine of gentle gymnastic exercises can contribute to maintaining better physical fitness. Video action recognition models can be employed by virtual or robotic coaches to assist older adults in maintaining an active and healthy lifestyle. To assess the effectiveness of our approach, we trained and tested two state-of-the-art video action recognition models on datasets that we collected and generated. Initially, we gathered a small real-life video dataset in our laboratory, which we subsequently expanded by generating a larger synthetic dataset using our Synthetic Video Generator. In the remaining part of this section, we will provide details about both the collected and generated datasets, as well as the action recognition models employed in our tests.

A. COLLECTED DATASET

We gathered 23 students from the Department of Mathematics and Computer Science at the University of Cagliari to collect two different sets of videos. The first set was recorded using the Intel RealSense D455 camera, while the second one used the onboard camera of the Xiaomi Poco3 cellphone. Each student performed each of the 11 exercises ten times, and we captured a video for each repetition. After reviewing the video data, we removed a few repetitions that were not captured properly, slightly reducing the size of the datasets. Informed consent of recording and storing those videos was obtained by all of them.

The first twenty students were recorded in a controlled environment inside our lab. The position of the RealSense

[3]https://github.com/nigno17/Synthetic-video-generator

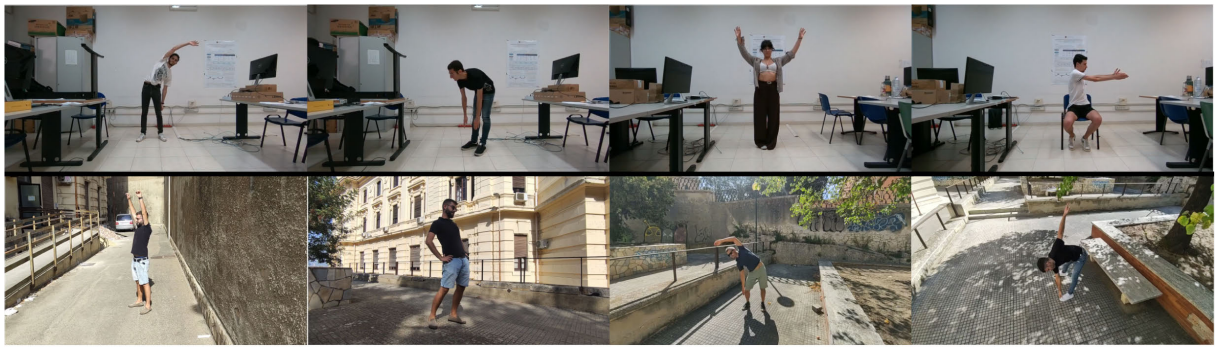


FIGURE 3. Examples of frames collected in real-life. Upper row: frames from training and validation datasets collected in the lab. Bottom row: frames from the outdoor test dataset.

camera remained fixed in front of the subjects and was not changed throughout the entire recording session. This dataset was split to create a training set from the first 15 students' recordings and a validation set from the last five students. See the upper row of Figure 3 for some example frames extracted from the training and validation set videos.

The video recordings of the last three students were used to create our test set. Videos in this dataset were collected outdoors, in the courtyard of our department, under variable light conditions, in multiple spots, and with variable camera positions. This dataset aimed to emulate a more realistic setup. The bottom row of Figure 3 displays some example frames extracted from this dataset.

To summarize, we divided our data into three datasets:

1) **Collected training set:** This dataset, used for training, contains 1647 videos. Each video corresponds to one repetition of one of the 11 exercises executed by one of the first 15 students recorded in the lab. Each exercise was repeated roughly 10 times by each student.
2) **Validation set:** This dataset, used for validation, contains 539 videos. Each video corresponds to one repetition of one of the 11 exercises executed by the remaining five students recorded in the lab. Each exercise was repeated roughly 10 times by each student.
3) **Test set:** This dataset, used for testing, contains 329 videos. Each video corresponds to one repetition of one of the 11 exercises executed by one of the three students recorded outdoors. Each exercise was repeated roughly 10 times by each student.

Training, Validation, and Test collected datasets are accessible online through the provided link in the footnote.[4]

B. GENERATED DATASET

To test the capability of our Synthetic Data Generator, we generated a synthetic video dataset. We extracted the body motions from all the videos in the "collected training set" using the Mediapipe pose landmark detection model. We then imported those body motions as Unity animation clips into our synthetic video generator via a C# script. We generated a dataset of 5000 videos displaying a human avatar performing one of the 11 exercises based on the imported animations. Each video was generated by randomizing avatar appearance (age, ethnicity, height, gender, body mass, and clothes), background images, scene illumination (color, intensity, and direction), animation speed, and camera position and orientation. Figure 4 displays some example of generated frames.

The generated data was used to create two new training sets:

1) **Generated training set:** This training contains 5000 videos and includes all the videos generated with the Synthetic Data Generator.
2) **Augmented training set:** This training set contains 6647 videos. It combines the 5000 videos generated with the Synthetic Dideo Generator with the 1647 videos collected in the lab for the "collected training set".

C. VIDEO ACTION RECOGNITION MODELS

To evaluate our datasets, we opted to employ two cutting-edge models: one based on Convolutional Neural Networks (CNNs), known as I3D, and the other utilizing Vision Transformer (ViT) architecture, referred to as Timesformer. We conducted our experiments using MMAction2 [53], an open-source PyTorch toolkit designed for video analysis. MMAction2 offers support for a wide range of models tailored for various video comprehension tasks, including action recognition, skeleton-based action recognition, spatio-temporal action detection, and temporal action localization. Additionally, it features an intuitive configuration script system for dataset management, model parameter configuration, and training and testing procedures. For both I3D and Timesformer, we employed pretrained weights trained on the ImageNet dataset.

[4]https://drive.google.com/drive/folders/1GzkfOD9byPzOhIPMceiBB6DRdCHRX4Ud?usp=drive_link

FIGURE 4. Examples of generated frames for our synthetic dataset. Please notice the range of scene randomization such as background images, illumination, avatar model, camera position and performed action.

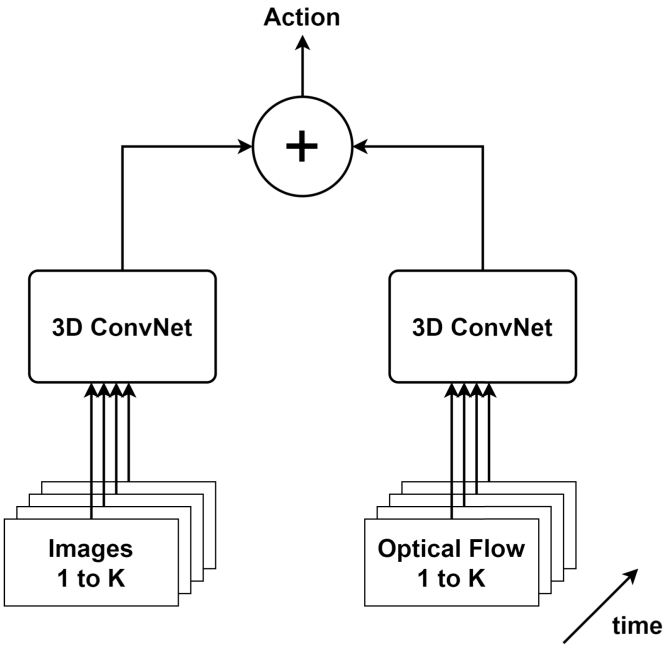


FIGURE 5. Two-Stream Inflated 3D-ConvNet (I3D) model. Carreira and Zisserman [22].

The Two-Stream Inflated 3D ConvNet (I3D) [22] is a video action recognition model based on 3D CNNs. I3D addresses the temporal aspect in videos through two primary mechanisms. Firstly, it employs a two-stream architecture to process both the actual RGB frames and a stream of optical flow (OF) maps using two identical CNN replicas. Secondly, it inflates the kernels and pooling layers of the two CNNs making them three dimensional. In the final model, the first stream takes a sequence of temporal RGB frames as input, while the second processes a sequence of extracted OF maps (as depicted in Figure 5). The inflated 3D CNNs in the I3D architecture are built upon the foundation of the ResNet50 network [54].

The Timesformer architecture [23] is an extension of the ViT model tailored for video action recognition. The attention model in Timesformer not only encompasses the spatial domain but also extends its influence over the temporal domain. The input to the Timesformer model consists of a sequence of RGB frames, with each frame divided into N non-overlapping patches. The self-attention mechanism, known as Divided Space-Time Attention, initially calculates the temporal attention between a patch and all corresponding spatial patches in other frames. Subsequently, it computes the spatial attention for the resulting temporal encoding with respect to all other patches within the analyzed frame (as shown in Figure 6).

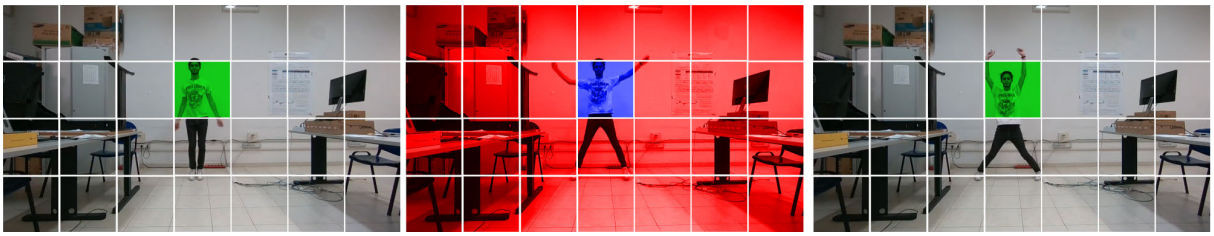


FIGURE 6. Example of the Divided Space-Time Attention model on a sequence of 3 frames. First, the temporal attention is calculated (green patches). Second, the spatial attention is calculated (red patches). Bertasius et al. [23].

TABLE 1. I3D validation and test accuracy results on our three training sets.

I3D				
	Validation Accuracy		Test Accuracy	
Traning set	Top-1	Top-5	Top-1	Top-5
Generated	0.748	0.996	0.733	**0.994**
Collected	0.920	0.996	0.191	0.571
Augmented	**0.931**	**1.000**	**0.833**	**0.994**

V. RESULTS

In this section, we present the outcomes of our tests. Our experimental setup was deliberately designed to replicate a real-world scenario characterized by the scarcity of data and the challenges of generalizing across highly variable conditions. In practical applications, access to large, well-curated datasets is often unrealistic, particularly when dealing with real-life, in-the-wild environments.

Our primary objective is to address this gap by demonstrating the effectiveness of our data augmentation approach in simulating the high variability of real-world test sets, which typically differ significantly from controlled training environments. In this specific case, the model is trained on a limited, biased dataset and tested on data recorded in uncontrolled, diverse conditions. The performance of any model under such constraints is bound to be lower than in ideal scenarios, but this limitation is what makes our proposed solution relevant. The gap between training and test conditions represents the challenge our augmentation technique aims to mitigate.

While additional experiments might yield further insights, our approach mirrors real-world constraints where the luxury of retraining or collecting more comprehensive data is rarely an option. Instead, the focus of our work lies in leveraging synthetic data to address these practical limitations and improve generalization.

We trained both the I3D and Timesformer models on the three training sets that were previously collected and generated: the collected training set, the generated training set, and the augmented training set. Following the training phase, we evaluated each model on both the validation and test sets described in the preceding section. Our evaluation focused on two key metrics: Top-1 and Top-5 accuracy. All tests were conducted on a laptop equipped with an AMD Ryzen 9 5900HX CPU and an NVIDIA GeForce RTX 3080 Laptop GPU. In each experiment, we trained the models, which were pretrained on ImageNet, for 15 epochs. Each frame was resized, with the smaller dimension reduced to 256 pixels, and central cropped to obtain 224×224 images. The ImageNet normalization was then applied to each cropped frame. For both networks, we utilized the stochastic gradient descent (SGD) optimizer with a learning rate of 0.005, a momentum of 0.9 and a weight decay of 0.0001.

A. I3D TESTS

Table 1 summarizes the validation and test Top-1 and Top-5 accuracy achieved by the I3D model, trained on our three training sets: the synthetic generated dataset, the lab-collected dataset, and the augmented dataset containing data from both sources. Looking at the validation results, the networks trained on the collected and augmented datasets outperformed of almost a 20% the model trained on the generated dataset in terms of Top-1 accuracy, with the latest achieving slightly better results. This outcome was expected, given that the validation set was captured under conditions identical to the collected training set, except for different subjects (same location and camera position). Nonetheless, the diversity of the generated training set enabled the model trained on it to achieve a remarkable 75% Top-1 accuracy on the validation set. Notably, all trained networks achieved nearly 100% Top-5 accuracy, indicating that even in cases where the first prediction was incorrect, they assigned high probabilities to the correct action.

The results on the test set diverged significantly from those on the validation set. The test set was collected under entirely different conditions from the collected training and validation sets. All videos in the test set were recorded outdoors in various locations within the university courtyard, using a different camera with varying subject-to-camera positions for each video. Consequently, the Top-1 accuracy of the model trained on the collected training set dropped significantly to 19%, rendering its predictions unreliable. In contrast, the models trained on the generated and augmented training sets maintained Top-1 accuracies around 73% and 83% respectively. Notably, the performance on the test set compared to the validation set dropped by only 2%

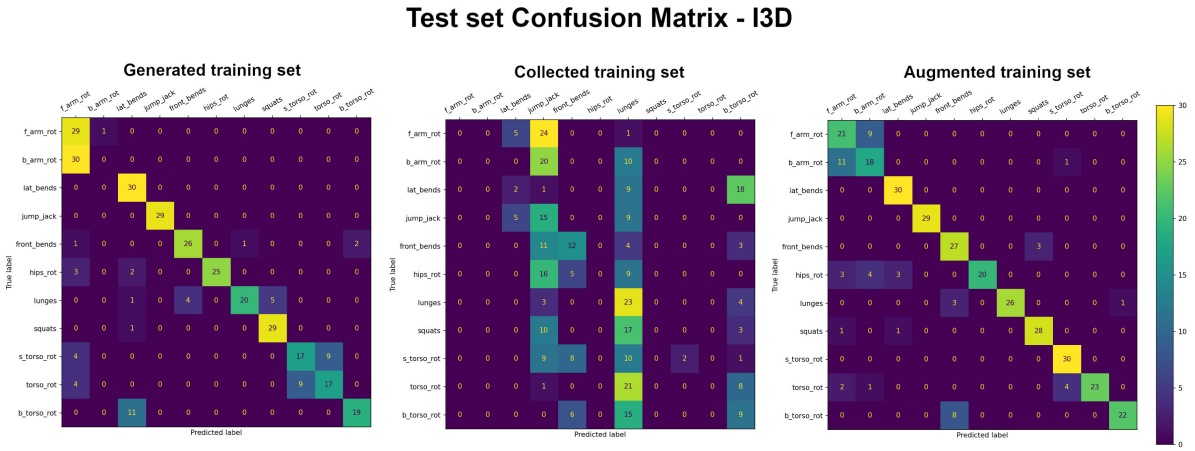


FIGURE 7. Confusion matrices obtained by running the I3D model on the test set, trained on the generated, collected, and augmented training sets.

when using the generated set for training, indicating that the generated videos are varied enough to cover the data distribution of both the validation and test sets. However, training with the augmented set resulted in an 10% drop, decreasing from 93% on the validation set to 83% on the test set. This decline was attributed to the inclusion of the biased collected data in the augmented training set. The Top-5 accuracy results exhibited a similar trend to the Top-1, with the model trained on the collected training set experiencing a drop from 99% to 57%, while the models trained on the generated and augmented sets maintained Top-5 accuracies of approximately 100%.

To gain deeper insights into the classifiers' behavior for each class, conducting a per-class analysis of the test set results is essential. Figure 7 illustrates the confusion matrices obtained by running the three versions of the I3D model on the test set. The left matrix corresponds to the model trained on the generated training set, the center one to the model trained on the lab-collected training set, and the right one to the model trained on the augmented training set. Consistent with the overall accuracy results, the models trained on the generated and augmented sets demonstrate proficiency in correctly classifying most of the classes. However, when exclusively using generated synthetic data for training, the I3D model encounters challenges in distinguishing between forward and backward arm rotations. It tends to classify all clips from these two classes as forward arm rotations. This difficulty arises due to the similarity between the two exercises, sharing the same body position with the only distinction being the direction of arm rotation. Additionally, the skeletal poses extracted from videos, utilized to generate the synthetic dataset, can introduce noise, contributing to the ambiguity in the generated clips. A similar, albeit less pronounced, issue is observed for standing and seated torso rotations. The absence of a chair in the generated clips for the seated torso rotations class makes it more challenging to differentiate from the standing version of the exercise.

TABLE 2. Timesformer validation and test accuracy results on our three training sets.

Timesformer				
	Validation Accuracy		Test Accuracy	
Traning set	Top-1	Top-5	Top-1	Top-5
Generated	0.768	0.979	0.812	**1.000**
Collected	**0.892**	**0.994**	0.541	0.912
Augmented	0.870	0.989	**0.824**	**1.000**

Notably, training the I3D model on the augmented set, which combines both generated and lab-collected data, alleviates these issues. In the confusion matrix on the right-hand side of Figure 7, forward and backward arm rotations, as well as standing and seated torso rotations, are better classified. The increased fidelity of avatar motions and the inclusion of the chair for the seated exercises in the collected data contribute to improved performance in these classes. Conversely, the network trained solely on collected data exhibits a complete failure on the test set. The lack of diversity in the lab-collected data prevents the model from generalizing effectively, as evidenced by the sparsely distributed confusion matrix.

B. TIMESFORMER TEST

The Timesformer model's results mirrored those of the I3D model, as shown in Table 2, displaying Top-1 and Top-5 accuracy results for both validation and test sets. The format of the table aligns with that of Table 1. For the validation set, the Timesformer model achieved commendable results when trained on all three training sets, with Top-1 accuracy ranging from 77% to 89% and Top-5 accuracy consistently reaching 98% or 99%. Similar to the I3D case, the model trained on the collected and augmented datasets exhibited over 10% better Top-1 accuracy compared to the model trained on the generated dataset. This advantage can be attributed to the

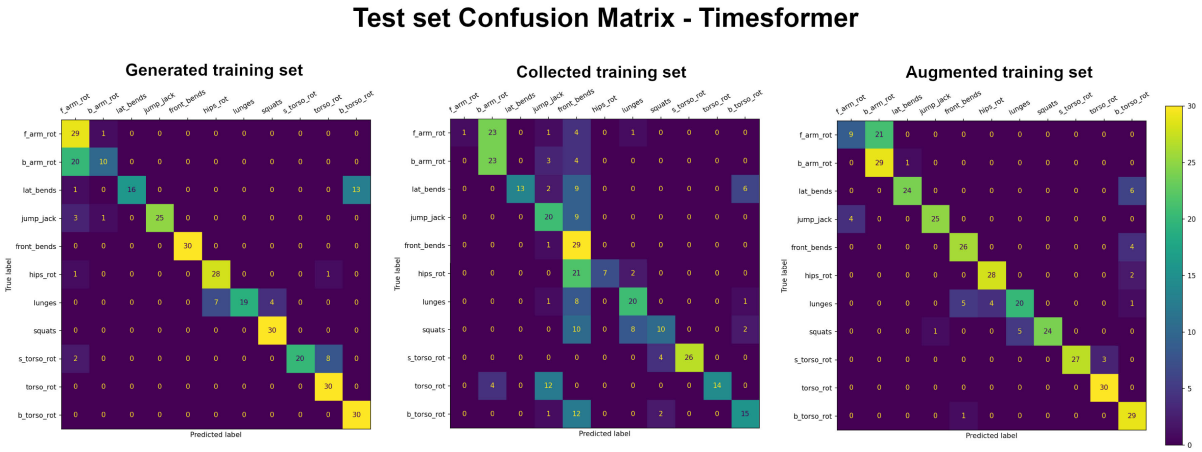


FIGURE 8. Confusion matrices obtained by running the Timesformer model on the test set, trained on the generated, collected, and augmented training sets.

identical collection conditions of the lab-collected training set and the validation set.

However, the drop in Top-1 accuracy on the test set was less pronounced for the Timesformer model compared to the I3D model when trained on the collected set, achieving an accuracy of 54%. Yet, the model's performance still decreased significantly compared to the results on the validation set. Remarkably, due to the greater variability in the generated data compared to the collected data, the model trained on the generated datasets improved its Top-1 accuracy on the test set, achieving 81%. This outcome suggests that the generated videos encompassed sufficient variability to match the data distribution of both the validation and training sets. Notably, the Timesformer model performed better on the test set than on the validation set, potentially indicating better generalization compared to the I3D model, which exhibited a slight tendency to overfit to the collected training set.

Figure 8 displays the confusion matrices obtained by the Timesformer on the test set. The results align with those of the I3D model shown in the previous subsection. Compared to the I3D, the Timesformer model appears to be less affected by overfitting and demonstrates better generalization on the test set. The challenges in classifying forward and backward arm rotations, as well as standing and seated torso rotations, observed in the I3D model trained solely on generated data are less pronounced when using the Timesformer. Furthermore, the model trained on lab-collected data, while still underperforming, exhibits better generalization on the test set compared to the I3D, evident in a confusion matrix with higher values along its diagonal.

VI. CONCLUSION

In this paper, we introduced a novel synthetic video generator developed in Unity for video action recognition. We harnessed this tool to create a video dataset tailored for the recognition of gentle gymnastic exercises. Our approach involved testing this generated dataset as the training set for two state-of-the-art video recognition models, I3D and Timesformer. We then compared the performance of models trained on the generated dataset with those trained on a dataset previously collected within the laboratory.

The results of our experiments demonstrated a significant enhancement in the generalization capabilities of both models when augmented with data generated by our tool. Furthermore, our findings illustrate that a dataset exclusively comprised of synthetic data is sufficient to train models for recognizing actions within videos collected in real-world settings, where conditions encompass variable locations, lighting, subjects, and camera positions.

While the results presented in this paper are promising, we are eager to further test and expand our model to enable more comprehensive comparisons with state-of-the-art methods. The first step is to upgrade our Unity-based synthetic generator to create videos featuring multiple actors and human interaction. Adding this feature will allow us to make a more thorough comparison with other models, as many of the benchmark datasets used by leading state-of-the-art works contain human interaction videos with multiple actors (e.g., UCF101 [43] and HMDB-51 [42] in [39], and NTU RGB+D [55] in [38]). Another interesting improvement to explore is the inclusion of external objects in the scene to better represent actions involving object interaction. We are developing a procedural method to incorporate handheld objects (e.g., cups, cellphones, laptops) and interactive objects (e.g., chairs, tables, doors).

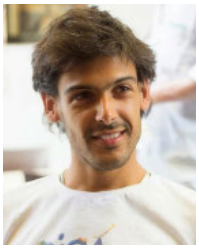
To facilitate further research and experimentation, we have made both our Unity synthetic video generator and the collected datasets available online. Interested researchers are encouraged to refer to Section III and Section IV

for access to these valuable resources, enabling the broader research community to leverage them in their work.

REFERENCES

[1] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019.
[2] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
[3] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés, "Video transformers: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12922–12943, Nov. 2023.
[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[5] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[7] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.
[8] H. Guan and M. Liu, "Domain adaptation for medical image analysis: A survey," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 3, pp. 1173–1185, Mar. 2022.
[9] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[10] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2351–2377, Mar. 2022.
[11] E. Goceri, "GAN based augmentation using a hybrid loss function for dermoscopy images," *Artif. Intell. Rev.*, vol. 57, no. 9, p. 234, Aug. 2024, doi: 10.1007/s10462-024-10897-x.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.
[13] E. Goceri, "Comparison of the impacts of dermoscopy image augmentation methods on skin cancer classification and a new augmentation method with wavelet packets," *Int. J. Imag. Syst. Technol.*, vol. 33, no. 5, pp. 1727–1744, Sep. 2023, doi: 10.1002/ima.22890.
[14] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
[15] E. Goceri, "Image augmentation for deep learning based lesion classification from skin images," in *Proc. IEEE 4th Int. Conf. Image Process., Appl. Syst. (IPAS)*, Dec. 2020, pp. 144–148.
[16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," 2018, *arXiv:1809.10790*.
[17] Unity Technologies. *Unity Homepage*. Accessed: Oct. 5, 2022. [Online]. Available: https://unity.com/
[18] E. Games. *Unreal Engine Homepage*. Accessed: Oct. 5, 2022. [Online]. Available: https://www.unrealengine.com/en-U.S./
[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 23–30.
[20] T. To, J. Tremblay, D. McKay, Y. Yamaguchi, K. Leung, A. Balanon, J. Cheng, W. Hodge, and S. Birchfield, "NDDS: NVIDIA deep learning dataset synthesizer," in *Proc. CVPR Workshop Real World Challenges New Benchmarks Deep Learn. Robotic Vis.*, 2018, pp. 18–22. [Online]. Available: https://github.com/NVIDIA/Dataset_Synthesizer
[21] H. Hwang, C. Jang, G. Park, J. Cho, and I.-J. Kim, "ElderSim: A synthetic data generation platform for human action recognition in eldercare applications," *IEEE Access*, vol. 11, pp. 9279–9294, 2023.
[22] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6299–6308.
[23] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. ICML*, vol. 2, no. 3, 2021, p. 4.
[24] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," 2018, *arXiv:1806.11230*.
[25] K. Jia and D.-Y. Yeung, "Human action recognition using local spatio-temporal discriminant embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 290, Jun. 2008, pp. 1–8.
[26] C. Yuan, B. Wu, X. Li, W. Hu, S. Maybank, and F. Wang, "Fusing $\mathcal{R}$ features and local features with context-aware kernels for action recognition," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 151–171, Jun. 2016.
[27] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," 2016, *arXiv:1609.08675*.
[28] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.
[29] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva, "Moments in time dataset: One million videos for event understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 502–508, Feb. 2020.
[30] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[31] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 568–576.
[32] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4694–4702.
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[35] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6836–6846.
[36] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4651–4664.
[37] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 336–345.
[38] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," *Int. J. Comput. Vis.*, vol. 129, no. 7, pp. 2264–2287, Jul. 2021.
[39] C. R. De Souza, A. Gaidon, Y. Cabon, and A. M. López, "Procedural generation of videos to train deep action recognition networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 2594–2604.
[40] N. Cauli and D. R. Recupero, "Survey on videos data augmentation for deep learning models," *Future Internet*, vol. 14, no. 3, p. 93, Mar. 2022.
[41] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2929–2936.
[42] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
[43] K. Soomro, A. R. Zamir, and M. Shah, "UcF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[44] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman, "A short note on the Kinetics-700–2020 human action dataset," 2020, *arXiv:2010.10864*.
[45] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action Mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[46] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1725–1732.
[47] S. Singh, S. A. Velastin, and H. Ragheb, "MuHAVi: A multicamera human action video dataset for the evaluation of action recognition methods," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2010, pp. 48–55.
[48] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.
[49] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 249–257, Nov. 2006.
[50] Unity Technologies. *Unity Perception Package Github Page*. Accessed: Oct. 10, 2023. [Online]. Available: https://github.com/Unity-Technologies/com.unity.perception
[51] Unity Technologies. *Unity Synthetic Humans Github Page*. Accessed: Oct. 10, 2023. [Online]. Available: https://github.com/Unity-Technologies/com.unity.cv.synthetichumans
[52] Google. *Mediapipe Homepage*. Accessed: Oct. 10, 2023. [Online]. Available: https://developers.google.com/mediapipe
[53] OpenMMLab. *MMAction2 Github Page*. Accessed: Oct. 27, 2023. [Online]. Available: https://github.com/open-mmlab/mmaction2
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[55] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.



NINO CAULI received the M.Sc. degree in computer science from the University of Pisa, Pisa, Italy, in 2010, and the Ph.D. degree (cum laude) in biorobotics from the BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, in 2014. He is currently a Researcher with the Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy. He has collaborated on several EU Projects, such as RoboSom and HumanBrain projects. His current research interests include deep neural networks, machine learning, computer vision, internal models, predictive controllers, and bioinspired robotics.

DIEGO REFORGIATO RECUPERO received the Ph.D. degree in computer science from the University of Naples Federico II, Italy, in 2004. From 2005 to 2008, he was a Postdoctoral Researcher with the University of Maryland, College Park, MD, USA. He has been a Full Professor with the Department of Mathematics and Computer Science, University of Cagliari, Italy, since February 2022. He co-founded six companies within the ICT Sector and is actively involved in European projects and research (with one of his companies he won more than 40 FP7 and H2020 projects). His current research interests include sentiment analysis, semantic web, natural language processing, human–robot interaction, financial technology, and smart grids. He is the author of more than 200 conference and journal papers in these research fields, with more than 2800 citations. He won different awards in his career (such as the Marie Curie International Reintegration Grant, the Marie Curie Innovative Training Network, the Best Researcher Award from the University of Catania, the Computer World Horizon Award, the Telecom Working Capital, the Startup Weekend, and the Best Paper Award).

Open Access funding provided by 'Università degli Studi di Cagliari' within the CRUI CARE Agreement