DIY Human Action dataset Generation

Mehran Khodabandeh *
mkhodaba@sfu.ca
Simon Fraser University

Hamid Reza Vaezi Joze *
Ilya Zharkov
Vivek Pradeep
{hava,zharkov,vpradeep}@microsoft.com
Microsoft

Abstract

The recent successes in applying deep learning techniques to solve standard computer vision problems has aspired researchers to propose new computer vision problems in different domains. As previously established in the field, training data itself plays a significant role in the machine learning process, especially deep learning approaches which are data hungry. In order to solve each new problem and get a decent performance, a large amount of data needs to be captured which may in many cases pose logistical difficulties. Therefore, the ability to generate de novo data or expand an existing dataset, however small, in order to satisfy data requirement of current networks may be invaluable. Herein, we introduce a novel way to partition an action video clip into action, subject and context. Each part is manipulated separately and reassembled with our proposed video generation technique. Furthermore, our novel human skeleton trajectory generation along with our proposed video generation technique, enables us to generate unlimited action recognition training data. These techniques enables us to generate video action clips from an small set without costly and time-consuming data acquisition. Lastly, we prove through extensive set of experiments on two small human action recognition datasets, that this new data generation technique can improve the performance of current action recognition neural nets.

1. Introduction

After significant successes in face detection, face recognition and object detection commonly used in our daily life, computer vision researchers are now aiming at understanding video which is one dimension more difficult. These successes rely on advanced machine learning techniques and training data which require computational power, mainly deep networks. Hence, the process of data acquisition may be as vital as the technique used. Large datasets, such as a million object and animal photos [26], hundreds of thousands of faces [22] or millions of scenes [29], enables complex neural networks to train successfully. However, similar results can never be achieved through small datasets manually captured by researchers themselves. Video datasets or specifically human action datasets are more difficult to compile. A few common scenarios to generate a human action dataset are as follows: (1) asking subjects to do a series of actions in front of a camera (2) collecting and labeling existing videos from the internet or crowd sourcing [49] (3) 3D video synthesizing[58, 5]. The first scenario is not scalable considering the number of subjects and the limitations imposed by the capturing environment. These types of datasets are not common anymore due to their small size. Some examples of the second scenario are UCF 101 [51] containing 101 actions of thousands of online clips, Hollywood2 [32] containing 12 actions in ~3000 clip extracted from movies and the kinetics [21] including 400 actions from hundreds of thousands of YouTube videos. Although

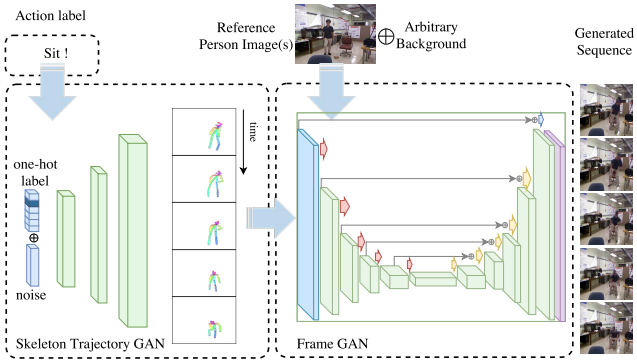


Figure 1: Our algorithm takes as input an action label, a set of reference images and an arbitrary background. The output is a generated video of the person in the reference image performing a given action. We approached this problem in two stages. Firstly (left side) a generative model trained on a small *labeled* dataset of skeleton trajectories of human actions, generates a sequence of human skeletons conditioned on the action label. Secondly (right side), another generative mode trained on an *unlabeled* set of human action videos, generates a sequence of photo-realistic frames conditioned on the given background, generated skeletons, and the person's appearance given in the reference frames. This produces an arbitrary number of human action videos.

*Equal contribution.

these datasets are very useful to benchmark the accuracy of different algorithms, the clips or actions are not necessarily useful for real world action recognition tasks such as security surveillance cameras, sport analysis, smart home devices, health monitoring etc, as each scenario has different settings and sets of actions. The drawback of the last scenario, which is more recent and seems more promising requires MoCap data.

In this paper, we've introduced a novel way to partition an action video clip into action, subject and context. We showed that we can manipulate each part separately and assemble them with our proposed video generation model into new clips. The actions are represented by a series of skeletons, the context is an still image or a video clip, and the subject is represented by random images of the same person. We can change an action by extracting it from an arbitrary video clip, generate it through our proposed skeleton trajectory model, or by applying perspective transform on existing skeleton. Additionally, we can change the subject and the context using arbitrary video clips, enabling us to arbitrarily generate action clips. This is particularly useful for action recognition models which require large datasets to increase their accuracy. With the use of a large unlabeled data and a small set of labeled data, we can synthesize a realistic set of training data for training a deep model.

We called it DIY (do it yourself) because we can eventually build our own dataset from a small one. Similar to actual data collection, not only we can add a new person or action to the dataset, but also internally expand the dataset or capture the same data from different angles with very little time and effort.

Lastly, to quantitatively evaluate our data generation technique, we applied it to UT Kinects [65] a human action dataset comprised of 10 actions in 200 video clips. We generated new video clip types by adding new subjects or actions or by expanding current action and subjects. It is shown that generated data along with the existing data, can improve the performance of well-performed video representation networks: I3D [4] and C3D [54] on action recognition task. For further investigation, we applied our method and action recognition task to actions with two persons in SUB interact [69] datasets. The outline of this paper is as follows. In §2 we've described related works in action recognition, data augmentation and video generative model. Section 3 introduces our video generation methods as well as skeleton trajectory generation methods with samples and use cases. In §4, we've discussed the datasets and action recognition methods used to evaluate our work. In §5 we've presented the extensive experimental data backing our claims. Our paper is concluded in §6.

2. Related Works

2.1. Action Recognition

Human action recognition has drawn attention for some time. Before deep learning era of computer vision, many researchers tried to inflate successful 2D features or descriptors in order to solve this problem such as 3d SIFT [47], 3d bag of features [27], dense trajectories [62], tracking [41, 40], and automatic target recognition [42]. Please refer to [36] for a comprehensive survey of these types of algorithms.

Deep learning networks significantly outperformed transitional approaches and are therefore the focus of this paper. Unlike image representation network architecture, the video representation networks haven't had satisfactory advances. There have been different approaches to this problem. Some used the convolution and layers in 2D (image-based) [8, 68] while some used 3D (video-based) kernels [16, 54, 4]. Input to the networks could be just RGB video [54] while optical flow could be used as an additional input [10, 4]. Information could propagate across frames either through LSTMs [8, 68] or feature aggregation [19].

Data Augmentation Using synthetic data or data warping for training classifiers has been proven effective [26, 71, 50]. Sato *et al*. [45] proposes a method for training a neural network classifier using augmented data. Wong *et al*. [64] thoroughly investigated the benefits of data augmentation for classification tasks. In action recognition tasks, data is usually very limited [24], since collecting and annotating videos [23] is difficult. Although one can use our algorithm for data augmentation by generating videos varying in background, human appearance, and type of actions, this is not the purpose of our work. Unlike data augmentation that is limited to manipulating data, our method is capable of generating new data with new content and visual features.

2.2. Video Generative Models

Video generation has posed as a challenge for a number of years. The early work in the field focused on generating texture [9, 53, 63]. In recent years with the success of generative models in image generation such as GANs [12], VAEs [25, 38], Plug&Play Generative Networks [34], Moment Matching Networks [28], and. PixelCNNs [57], a new window of opportunity has opened towards generating videos using generative models. In this paper, we use GANs to generate human skeleton trajectories and realistic video sequences. GAN consists of a discriminator and a generator, trained in a 2-player zero-sum game. Although GANs have shown promising results on image generation [7, 37, 70, 31, 30], they have proven to be difficult to train. To address this issue, Arjovsky *et al*. [1] proposed Wassertein GAN to combat mode collapse with more stability. Salimans *et al*. [44] introduced several tricks for training

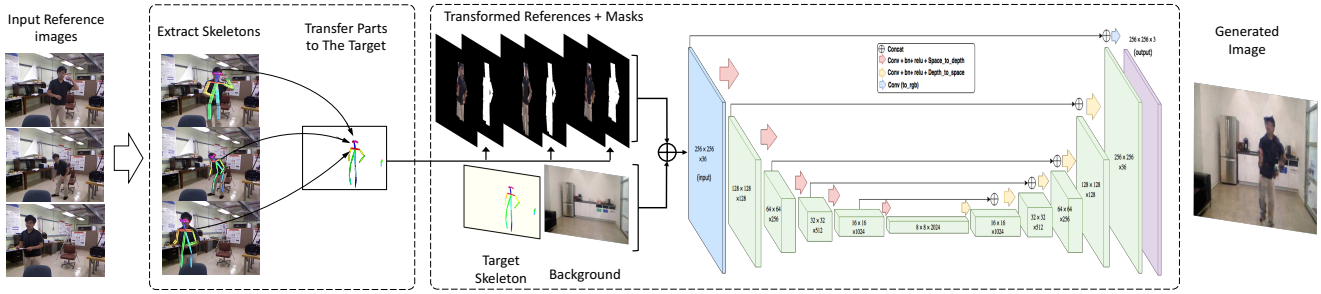


Figure 2: Structure of the network. On the left side "generator network" takes as input background, target skeleton, and the transformed reference images to the target skeleton along with their masks. On the right side "discriminator" takes as input generated image or ground truth and outputs "fake" or "real".

GANs. Karras *et al*. [20] proposed a novel method for training GANs through progressively adding new layers. Ronneberger *et al*. [39] proposed U-Net, a convolutional network for segmentation.

GANs have previously been used for video generation. There are two lines of work in video generation. First is video prediction where given the first few frames of a video, the goal is to predict the future frames. Several papers focus on producing pixel values conditioned on the past observed frames [67, 52, 35, 33, 18, 66, 59]. Another group of papers aimed at reordering the pixels from the previous frames to generate the new ones [56, 11].

In the second line of work, the goal is to generate a sequence of video frames conditioned on label, single frame, etc. Early attempts assumed video clips to be fixed length and embedded in a latent space [60, 43]. Tulyakov *et al*. [55] proposed to decompose motion from content and generate videos using a recurrent neural net. Our work is different from [55] where their model learns motion and content in the same network whereas we separated them completely.Furthermore, [55] is not capable of generating complex human motions. Also filling gaps in the background initially blocked by the person in the input video is a difficult task for this method. On the other hand, our method handles these challenges by completely separating appearance, background, and motion. Our work is somewhat similar to [61], which does video forecasting using pose estimation, by modeling the movement of human using a VAE and then using a GAN to predict the pixel value of the future frames.

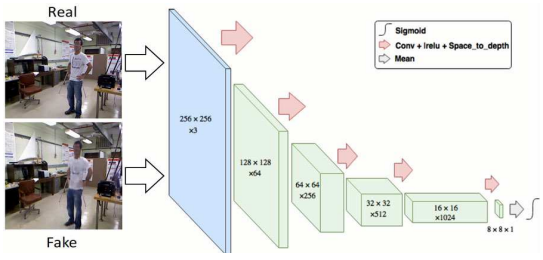


Figure 3: Architecture of the discriminator, D.

Our work lies in the "video generation" category where we focus on employing video generation techniques to generate human action videos. In our proposed method we completely separate background, skeleton motion, and appearance, allowing us to model frame generation and skeleton trajectory independently. So, one would require labeled data and the other can benefit from unlimited unlabeled human action videos available on internet, respectively.

3. Method

We define problem as follows; given an action label l a small set of reference images $I = \{I_1, ...I_k\}$ each containing a human subject from which a sequence of video frames is generated featuring a human with the same appearance as the human in the reference image set I performing an action l. Modeling the (human/camera) motion and generating photo-realistic video frames may be challenging but knowing the location/motion of human skeletons in each frame would simplify it. Hence, we subdivided the problem into two simpler tasks (inspired by [55, 59]).

- The first task comprised of the reference images I, background image B, and a sequence of target skeletons $S = [S_1, S_2, ..., S_n]$ employed to render photo-realistic video frames of the person in I moving according to S on background.
- The second task produced the target skeleton sequences for the first part. In another words, given action label l, a sequence of skeletons of a random person performing action l was generated.

By combining the two tasks, we created a novel algorithm that can generate arbitrary number of human action videos with varying backgrounds, human appearances, actions, and ways each action is performed.

3.1. Video Generation from Skeleton and Reference Appearance

In this section, we explain our algorithm used to generate a video sequence of a person based on given appear-

ance (I) and a series of target skeletons (S) in an arbitrary background(B). In our proposed model, we use GAN conditioned on the appearance, the target skeleton, and the background. Our proposed generator network works in a frame-by-frame fashion, where each frame is generated independently from others. We have tried using LSTMs and RNNs to take into account smoothness of the videos. However, our experiments show frames that are generated separately are sharper as RNNs/LSTMS may introduce blurriness to the generated frames.

Generator Input. Our generator network needs a reference image of the person in order to generate images of the same person with arbitrary poses/backgrounds. However, one reference image may not have all the appearance information due to occlusions in some poses (e.g. face is not visible when the person is not facing the camera). To overcome this issue to some extent, we provided multiple reference images of the person to the network. In both training and testing, these images were selected completely at random, so that network would be responsible for choosing the right pieces of appearance features from the set of input images. These images could be selected with a better heuristic to produce better results though this is not in the scope of this work.

The reference images were pre-processed before incorporation into the network. First we extracted the human skeleton from each reference image I_i (using [3]), then used an offline transform to map the RGB pixel values of each skeleton part from the image to the target skeleton. Also, a binary mask of where the transformed skeleton is located was created. All these images, $I^t = \{I_1^t, ..., I_k^t\}$, along with the background, B, and the target skeleton, S_i were stacked.

Conditional GAN. Inspired by pix2pix [15], we used a U-net style conditional GAN. The generator $G(C)$, is conditioned on the set of transformed images and corresponding masks, along with the background and target skeleton. The generator, G, maps $C = \{I_1^t, ..., I_k^t, B, S_i\}$ to the target frame Y, such that it fools the discriminator, $D(C, Y)$. The discriminator, $D(C, Y)$, on the other hand is trained to discriminate between real images and the fake images generated by G. The architecture of the discriminator is illustrated in Fig. 3. The pipeline and architecture of the generator G is illustrated in Fig. 2. Fig. 4a illustrates some of the results.

The objective function of GAN is expressed as:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{c,y\sim P_{data}(c,y)}[\log D(c, y)] + \mathbb{E}_{c\sim P_{data}(c),z\sim P_z(z)}[1 - \log D\big(c, G(c, z)\big)]$$

Following [15] we added an $L1$ loss to the objective function, which resulted in sharper generated frames.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{c,y\sim P_{data}(c,y),z\sim P_z(z)}[||y - G(c, z)||]$$

In initial experiments, we noticed that using only $L1$ loss and GAN loss is not enough as the output background would be sharp but the region that the target person is supposed to be was blurry. Subsequently, we introduced a "Regional L1 loss" with a larger weight as following,

$$\mathcal{L}_R(G) = \mathbb{E}_{c,y\sim P_{data}(c,y),z\sim P_z(z)} [||\text{masked}(y) - \text{masked}(G(c, z))||]$$

where "masked" masks out the region where the person was located. This mask was generated based on the target skeleton, S_i, using morphological functions (erode, etc.).

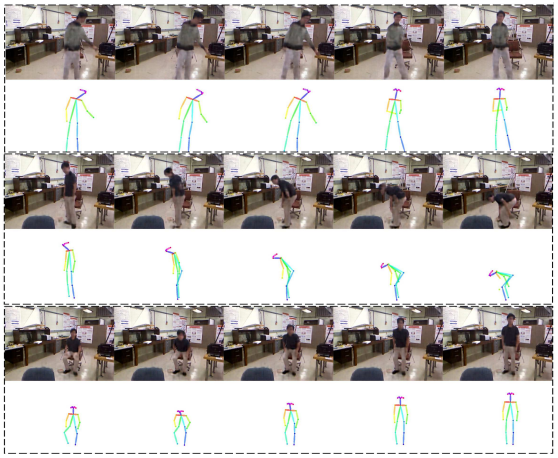
Our final objective is as follows:

$$\mathcal{L}(G, D) = \mathcal{L}_{GAN}(G, D) + \lambda\mathcal{L}_{L1}(G) + \beta\mathcal{L}_R(G)$$

where λ and β are weights of $L1$ and R regional losses (in our experiments $\beta > \lambda$). and the goal is to solve the following optimization problem.

$$G^* = \arg\min_G \max_D \mathcal{L}(G, D) \quad (1)$$

Multi-person Video Generation In a nutshell, our algorithm merges transformed images of a person on an arbitrary pose with an arbitrary background in a natural photo-realistic way. We managed to go beyond simple one per-



(a) UT dataset. Subjects from the same dataset.

(b) SBU dataset. None of the subjects exist in this dataset.

Figure 4: Generated images on two different datasets.

son human action videos and extended our method to *multi-person interaction videos* as well. For this purpose, we trained our model on a two person interaction dataset [69]. The only difference with single frame generation process is that in the pre-processing phase, for each person in the input reference image, we needed to know the corresponding skeleton in the target frame, we then transformed each person's body parts to his/her own body parts in the target skeleton. There are some challenges in this task such as occlusions in certain interactions (e.g. passing by, hugging, etc.). The dataset that we used contains these occlusions to some extent. Our method is able to handle relatively well some simple occlusions that occur in such interactions. We acknowledge that there is room for improvement in this area, but that would not fit in the scope of this work. Fig. 4b illustrates some of the generated videos.

3.2. Skeleton Trajectory Generation

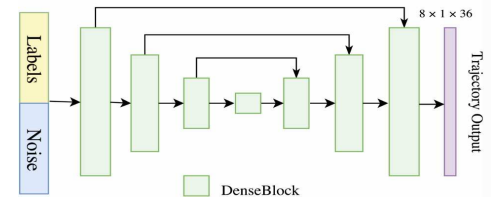
In the previous section, we explained how we designed a method that enables us to generate videos of an arbitrary person in any background based on any given sequence of skeletons. Although number of backgrounds and persons are unlimited, the number of labeled skeleton sequences are limited to the ones in the existing datasets. We propose a novel solution to this problem; using a generative model to learn the distribution of skeleton sequences conditioned on the action labels. This allows us to generate as many skeleton sequences as needed for the actions in the dataset. Fig. 6 shows a few sample generated skeleton sequences.

We used small datasets for training our model. However, due to the nature of the problem and the limited amount of data, generating long sequences of natural looking skeletons proved challenging. Thus we aimed at generating relatively short fixed-length sequences. Having said that, training GAN in such way is still prone to problems such as mode collapse, divergence, etc. In designing the generator and discriminator networks, we have taken into account these problems (e.g. introduced batch diversity in the discriminator, created multiple discriminators, etc.).

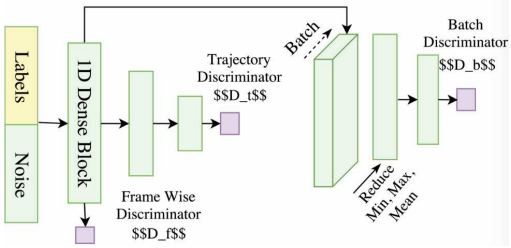
Skeleton Trajectory Representation. Each skeleton consists of 18 joints. We represented each skeleton with a 1×36 vector (a flattened version of 18×2 matrix of joints coordinates). We normalized the coordinates by dividing them by "height" and "width" of the original image.

Generator Network. We used a conditional GAN model to generate sequences of skeletal positions corresponding to different actions. Our generator has a "U" shape architecture where input consists of action label and noise, and output is a $8 \times 1 \times 36$ tensor representing a human skeleton trajectory with 8 time-steps.

Based on our results, providing a vector of random noise for each time step helps the generator to learn and generalize better. So the input noise, z, is a tensor with size



(a) Generator Network.



(b) Trajectory Discriminator Network. The discriminator is the sum of three discriminators illustrated in this figure: $D = D_f + D_t + D_f$.

Figure 5: Trajectory GAN network architecture.

$8 \times 1 \times 128$; drawn from a uniform distribution. The one-hot encoding of action label, l, is replicated and concatenated to the 3rd dimension of the z. The rest is a "U" shaped network with skip connections that maps the input (z, l) to a skeleton sequence S. Fig. 5a illustrates the network architecture. We also used Dense-net [13] blocks in our network.

Discriminator Network. Architecture of discriminator is three-fold. The base for discriminator is 1D convolutional neural net along the time dimension. In order to allow discriminator to distinguish "human"-looking skeletons, we used sigmoid layer on top of fully-convolutional net. To discriminate "trajectory", we used set of convolutions along the time with stride 2, shrinking output to one $1 \times 1 \times C$ containing features of the whole sequence. To prevent mode collapse, first we grouped fully convolutional net outputs across batch dimension.We then used min, max and mean operations across batch, and provided these statistical information to the discriminator. This method seems to provide enough information about distribution of values across batch and allows to change batch size during training. For detailed discriminator architecture see Fig. 5b.

Our objective function is:

$$\mathcal{L}_T(G, D) = \mathbb{E}_{l,s\sim P_{data}(l,s)}[\log D(l, s)] + \mathbb{E}_{l\sim P_{data}(l), z\sim P_z(z)}[1 - \log D\big(l, G(l, z)\big)]$$

where l and s are action label and skeleton trajectories, respectively. We aim to solve the following:

$$G^* = \arg\min_G \max_D \mathcal{L}_T(G, D)$$

In this work, we have shown that generative models can be adopted to learn human skeleton trajectories. We trained

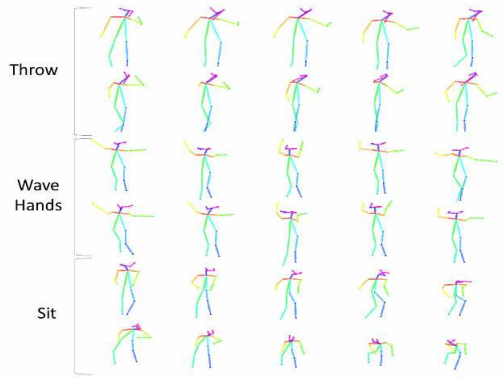


Figure 6: Samples of generated skeleton sequences, conditioned on action label (e.g. throwing, hand waving, sitting).

a Conditional GAN on a very small dataset (200 sequences) and managed to generate natural looking skeleton trajectories conditioned on action labels. This can be used to generate a variety of human action sequences that don't exist in the dataset. However, our work is limited to a fixed number of frames. Thus for future work, we'll work to improve our method so that it'll accommodate longer sequences varying in length. We also explained that in addition to the generated skeletons, we can also use real skeleton sequences from other sources (other datasets, current dataset but different subjects) to largly expand existing datasets.

4. Datasets and Action Recognition Methods

4.1. datasets

In this paper, we've claimed to expand small amount of action videos by addition of new generated videos. We targeted smaller action recognition datasets and expanded them to meet the large data load requirements of recent action recognition algorithms such as UCF 101 [51], the kinetics [21] or NTU RGB+D [48]. This eliminates the need for time and cost inefficient data acquisition processes.

UT Kinects [65]: One of the datasets wildly used in our experiments is UT Kinects which includes 10 action labels: Walk, Sit-down, Stand-up, Trow, Push, Pull, Wave-hand, Carry and Clap-hand. There are 10 subjects that perform each of these action twice in front of a rig of RGB camera and Kinect. Therefore in total they are 200 action clips of RGB and depth though depth is ignored. All videos are taken in office environment with similar lighting condition and the position of the camera is fixed.

For the training setup, 2 random subjects were left out (20%, used for testing) and the experiments were carried out using 80% of the subjects. The reported results are the average of six individual runs. The 6 train/test runs are constant throughout our experiment.

SUB Interact [69]: Since our methods work with multiple human subjects in a scene, we picked SUB Interact. It is a kinect captured human activity recognition dataset depicting two person interaction. It contains 294 sequences of 8 classes (Kicking, Punching, Hugging, Shaking-hand, Approaching, departing and Exchanging objects) with subject independent 5-fold cross validation. The original data includes RGB, depth and skeleton but we only use RGB for our purpose. We used a 5-fold cross validation throughout our experiments and reported the average accuracy.

KTH [46]: KTH action recognition dataset was commonly used at the early stage of action recognition. It includes 600 low resolution clips of 6 actions: Walk, Wavehand, Clap-hand, Jogging, running and boxing which are divided in train, test and validation. The first three action labels are shared with UT dataset while the last three are new. We used this dataset to add new action to UT dataset and for cross dataset evaluation.

4.2. Action Recognition Methods

We used the following deep learning networks which have previously shown decent performance on recent action recognition datasets.

Convolutional 3D (C3D) [54]: is a simple and efficient 3-dimensional ConvNet for spatiotemporal feature which shows decent performance on video processing benchmarks such as action recognition in conjunction with large amount of training data. We used their proposed network with 8 convolutional layers, 5 pooling layers and 2 fully connected layers with 16-frames of 112×112 RGB input. They released a network pre-trained on UCF Sport [51] which we used for our experiments aimed at training from scratch, denoted as C3D(p) vs. C3D(s). Unfortunately we can not couldn't converge the C3D when we trained from scratch on UT dataset but it converged successfully on SUB.

Inflated 3D ConvNets (I3D) [4] : is a more complex model which has recently been proposed as the state-of-the-art for action recognition task. It builds upon Inception-v1 [14], but inates their filters and pooling kernels into 3D. It is a two-steam network which uses both RGB and optical flow input with 224×224 inputs. We only used RGB for simplicity. They released a network pre-trained on ImgeNet [6] followed by the Kinetics [21]. We used this for our experiments aimed at training from scratch, denoted as I3D(p) vs. I3D(s).

We use data augmentation by translation and clipping as mentioned in [4] for all experiments. For training, we only used the original clips as test, making sure there was no generated clips with skeletons or subjects (subject pair) from test data in each run.

5. Experiments

So far, we have introduced our video generation method which enable us to generate new action clips for the action

recognition training process. In this section, we show different scenarios for generating new data and running experiments for each to see if adding the generated data to a training process can improve the accuracy of the action recognizer. We applied our proposed video generation models to all the experiments using skeletons. The skeletons were trained using data from UT and SBU datasets as well as 41 un-annotated clips (between 10 to 30 seconds) that we captured from our colleagues. For future works, we will train our model again using a large amount of data from web. But the time being, we are satisfied with the current model as higher resolution for action recognition is currently unnecessary. Our technique for generating new action video clips has the capacity of running experiments with numerous varying settings. Here, we show five experiments which may be quantitatively evaluated.

5.1. Generated Trajectory

The first experiments is a combination of our proposed video generation technique and skeleton trajectory generation. We generated around 200 random skeleton trajectories from action labels in UT dataset using the method mentioned in §3.2. Each of these skeleton trajectories generated a video by proposed video generation applied to a person in UT dataset, meaning our new dataset is doubled with half of it being the generated data. We then trained our model by I3D and C3D using training setting mentioned in §4.1. Table 1 shows about 3% improvement for I3D with and without training data as well as significant improvement (by 15%) for C3D network which is less complex.

Method	Org.	Org. + Gen.
I3D(s)	64.58%	67.50%
I3D(p)	86.25%	89.17%
C3D(p)	55.83%	70.83%

Table 1: Action recognition on UT dataset using original data compared to generated from scratch data with proposed method in §3.1 and §3.2

5.2. New Subjects

One common way to extend a video dataset is to invite new people to do a series of actions in front of a camera. Diversity [2] in body shape, cloths and behaviour will clearly help with the generalization of the ML methods. In this experiment, we aimed to virtually add new subject to the dataset. Thus, we collected a small unannotated clips from 10 distinct persons and fed them as new subjects into our proposed video generation method. For UT, each subject was replaced by a new one for all of his/her action which is similar to adding 10 new subjects to UT. The same was done with SUB to double the dataset, the only difference being the replacement each pair with a new subject pair. Figure 4b shows a few new subjects with their generated action videos from SBU dataset. The results have been presented in Table 2.

	UT		SBU	
	Org.	Exp.	Org	Exp.
I3D(s)	64.58%	67.08%	86.48%	91.23%
I3D(p)	86.25%	89.17%	97.30%	98.65%
C3D(s)	-	-	83.52%	87.00%
C3D(p)	55.83%	70.43%	92.02%	96.25%

Table 2: Performance comparison of multiple algorithms, trained on original data and additional subjects.

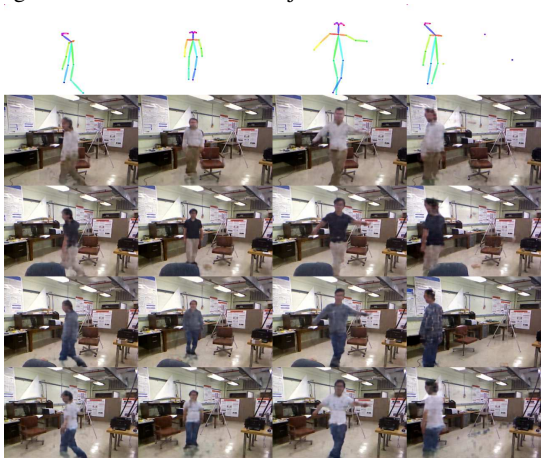


Figure 7: The screen shot of a video generated by UTK expansion. The first row shows skeleton clips extracted from an arbitrary action. Second to fourth rows show the generated video for subjects from different clip carrying out that specific action.

5.3. New Actions

In real computer vision problems, one might decide to add a new label class after the data collection process has been done. Adding a new label action to a valid dataset could cost the same as gathering a dataset from scratch as all the subjects are needed for re-acting that single action. As mentioned in §4.1, UT consists 10 action labels. In this experiment, we try to introduce new actions (i.e. running, jogging, and boxing) to UT dataset, which do not already exist. We used the skeleton data, which are extracted by OpenPose [3], from the training set of a third dataset, KTH [46]. We randomly picked 5 clips from each of these 3 actions and used all the subjects of UT to generate 150 new video clips. We then trained a new model using a pre-trained I3D network on the union of the original training data of UT and the newly generated data (150 clips). Since the KTH data is grey scaled images, we randomly grey scaled both the original and the generated training clips in the training phase. For each run, we found per class accuracy for UT test set (refer to §4.1 for explaining UT train/test) as well as KTH test sets. Table 3 shows average of the per class accuracy for both test sets. We may consider KTH test results

as a measure of cross dataset accuracy for walk, wave-hand and clap-hand. Our trained network on new action labels *boxing*, *running* and *jogging* achieved 72.14%, 44.44% and 63.20%, respectively. This indicates that the new actions in the dataset performed as good as the data captured by camera.

Action	UTK Test	Label	KTH Test
Walk	91.67%	Walk	67.18%
Wave-hand	100.0%	Wave-hand	58.59%
Clap-hand	91.67%	Clap-hand	28.90%
Push	33.33%	**Boxing**	**72.14%**
Pull	58.33%	**Running**	**44.44%**
Pick-up	100.0%	**Jogging**	**63.20%**
Sit-down	87.50%		
Stand-up	95.83%		
Threw	54.17%		
Carry	79.17%		

Table 3: Per class average accuracy for model trained by i3d using original training data from UT plus new action clip generated by our method using skeleton extracted from KTH training set.

5.4. dataset Expansion

So far, we've shown that using our proposed method we can generate video clips with any number of arbitrary action videos and subjects. In an action dataset with N subjects carrying out M distinct actions, there will be $M \times N$ video actions. when applied to our proposed method of action video generation, the N subjects and the $M \times N$ video actions will result in generation of $M \times N^2$ video actions comprising of $M \times N$ original videos while the rest is generated videos. This approach enabled us to expand UT Kinect dataset from 200 clips to 4000 clips and SUB Interact from 283 clips to 5943 using only the original dataset. We trained I3D and C3D using our expanded dataset as described in §4.1. Table 4 shows the result of this experiment.

	UTK		SBU	
	Org.	Exp.	Org	Exp.
i3d(s)	64.58%	69.58%	86.48%	93.54%
i3d(p)	86.25%	90.42%	97.30%	99.13%
c3d(s)	-	-	83.52%	86.03%
c3d(p)	55.83%	71.25%	92.02%	97.41%

Table 4: The comparison of dataset expansion by original data for UTK and SUB dataset.

Figures 7 shows an screen shot of the clips from UTK and SUB datasets. The first row shows skeleton clips extracted from an arbitrary action while rows 2-4 show the generated video for subjects from different clip performing that specific action.

5.5. Real World

In this section, we carried out 4 different experiments on 2 datasets for bench-marking. Although in all experiments, the generated data improved the network performance, we believe none of the experiments show the actual strength and convenience of our proposed methods in real world scenarios. In both datasets, as well as other commonly used small datasets, the environmental setup for data acquisition such as distance from camera view [17] and light condition were kept as uniformly as possible for both test and train video clips. This would be unattainable in real life data acquisitions. A way of overcoming this obstacle would be to collect diverse sets of data for strong neural network models. We've previously shown that by partitioning the video to action, subject and context allows us to easily manipulate the background or change the camera view. In this experiment, We applied perspective transform on skeleton while using diverse backgrounds. Although the model trained with these data did not outperform our previous experiments, a live demo showed it to be better for unseen cases, qualitatively. Figure 8 illustrates an input skeleton and its perspective transform as well as the generated clip.

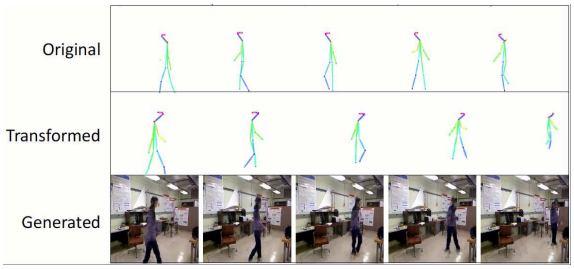


Figure 8: Perspective transform example.

6. Conclusion and Future Works

In this paper, we've introduced a novel way to partition an action video clip into action, subject and context. We showed that we can manipulate each part separately, reassemble them with our proposed video generation model into new clips and use as an input for action recognition models which require large data. We can change an action by extracting it from an arbitrary video clip, generate it through our proposed skeleton trajectory model or by applying perspective transform on existing skeleton. Additionally, we can change the subject and the context using arbitrary video clips.

For the future work, we will replace our 2d skeleton with 3d skeleton to achieve a 3d transformation and handle occlusions. Additionally, while our video generation technique demonstrated acceptable results for 255×255 images, we believe it can be extended even further to achieve higher resolution by feeding more unannotated data.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

[2] M. Bagheri, Q. Gao, S. Escalera, A. Clapes, K. Nasrollahi, M. B. Holte, and T. B. Moeslund. Keep it accurate and diverse: Enhancing action recognition performance by ensemble learning. In *CVPRW*, pages 22–29, 2015. 7

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4, 7

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*, 2017. 2, 6

[5] C. R. de Souza, A. Gaidon, Y. Cabon, and A. M. L. Peña. Procedural generation of videos to train deep action recognition networks. *CVPR*, 2017. 1

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 6

[7] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015. 2

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. 2

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 2

[11] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016. 3

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 6

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 4

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013. 2

[17] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1):172–185, 2011. 8

[18] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016. 3

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 6

[22] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, pages 4873–4882, 2016. 1

[23] M. Khodabandeh, Z. Deng, M. S. Ibrahim, S. Satoh, and G. Mori. Active learning for structured prediction from partially labeled data. *arXiv preprint arXiv:1706.02342*, 2017. 2

[24] M. Khodabandeh, A. Vahdat, G.-T. Zhou, H. Hajimirsadeghi, M. Javan Roshtkhari, G. Mori, and S. Se. Discovering human interactions in videos with limited data labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–18, 2015. 2

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1, 2

[27] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPRW*, pages 9–14. IEEE, 2010. 2

[28] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1718–1727, 2015. 2

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1

[30] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 2

[31] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 2

[32] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 1

[33] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 3

[34] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016. 2

[35] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*, pages 2863–2871, 2015. 3

[36] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 2

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[38] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, 2014. 2

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 3

[40] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6, 2017. 2

[41] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. *arXiv preprint arXiv:1711.10061*, 2017. 2

[42] A. Sadeghian, D. Lim, J. Karlsson, and J. Li. Automatic target recognition using discrimination based on optimal transport. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2604–2608. IEEE, 2015. 2

[43] M. Saito and E. Matsumoto. Temporal generative adversarial nets. *arXiv preprint arXiv:1611.06624*, 2016. 3

[44] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 2

[45] I. Sato, H. Nishimura, and K. Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015. 2

[46] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, volume 3, pages 32–36. IEEE, 2004. 6, 7

[47] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM, 2007. 2

[48] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016. 6

[49] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 1

[50] P. Y. Simard, D. Steinkraus, J. C. Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003. 2

[51] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 6

[52] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, pages 843–852, 2015. 3

[53] M. Szummer and R. W. Picard. Temporal texture modeling. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 823–826. IEEE, 1996. 2

[54] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 2, 6

[55] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*, 2017. 3

[56] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435*, 2017. 3

[57] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 2

[58] G. Varol, J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. *CVPR*, 2017. 1

[59] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 1(2):7, 2017. 3

[60] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016. 3

[61] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. *arXiv preprint arXiv:1705.00053*, 2017. 3

[62] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176. IEEE, 2011. 2

[63] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000. 2

[64] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: when to warp? In *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, pages 1–6. IEEE, 2016. 2

[65] L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, pages 20–27. IEEE, 2012. 2, 6

[66] T. Xue, J. Wu, K. Bouman, and B. Freeman. Probabilistic modeling of future frames from a single image. In *NIPS*, 2016. 3

[67] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2016. 3

[68] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. 2

[69] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPRW*, pages 28–35. IEEE, 2012. 2, 5, 6

[70] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016. 2

[71] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam. Learning classifiers from synthetic data using a multichannel autoencoder. *arXiv preprint arXiv:1503.03163*, 2015. 2