2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Vision-Based Gesture Recognition in Human-Robot Teams Using Synthetic Data

Celso M. de Melo[1], Brandon Rothrock[2], Prudhvi Gurram[1], Oytun Ulutan[3] and B.S. Manjunath[3]

Abstract— **Building successful collaboration between humans and robots requires efficient, effective, and natural communication. Here we study a RGB-based deep learning approach for controlling robots through gestures (e.g., "follow me"). To address the challenge of collecting high-quality annotated data from human subjects, synthetic data is considered for this domain. We contribute a dataset of gestures that includes real videos with human subjects and synthetic videos from our custom simulator. A solution is presented for gesture recognition based on the state-of-the-art I3D model. Comprehensive testing was conducted to optimize the parameters for this model. Finally, to gather insight on the value of synthetic data, several experiments are described that systematically study the properties of synthetic data (e.g., gesture variations, character variety, generalization to new gestures). We discuss practical implications for the design of effective human-robot collaboration and the usefulness of synthetic data for deep learning.**

I. Introduction

Robust perception of humans and their activities will be required for robots to effectively and efficiently team with humans in vast, outdoor, dynamic, and potentially dangerous environments. This kind of human-robot teaming is likely to be relevant across different domains, including space exploration [1], peacekeeping missions [2], and industry [3]. The cognitive demand imposed on humans by the sheer complexity of these environments requires multiple, effective, and natural means of communication with their robotic teammates. Here we focus on one visual communication modality: gestures [4]. Gestures complement other forms of communication - e.g., natural language - and may be necessary when (1) it is desirable to communicate while maintaining a low profile, (2) other forms of digital communication are not available, or (3) distance prevents alternative forms of communication. Since humans are likely to carry a heavy load and need to react quickly to emerging situations, it is important that gesture communication be untethered; thus, an approach based on data gloves is excluded [5]. The operational environment, moreover, is likely to be outdoors, thus, excluding the viability of depth sensors - such as the Kinect - which are known to have limitations in these settings. Therefore, in contrast to other work in gesture recognition [6] [7], we pursue a pure vision-based RGB solution to this challenge.

[1]Celso M. de Melo and Prudhvi Gurram are at the CCDC US Army Research Laboratory, Playa Vista, CA 90094, USA celso.miguel.de.melo@gmail.com, gurram_prudhvi@bah.com

[2]Brandon Rothrock perfomed this research at the NASA Jet Propulsion Laboratory, Pasadena, CA 91101 brandon.rothrock@gmail.com

[3]Oytun Ulutan and B.S. Manjunath are with the Electrical Engineering Department, UC Santa Barbara, Santa Barbara, CA 93106-9560, USA ulutan@ucsb.edu, manj@ucsb.edu

A. Activity & Gestures Recognition

Recognizing human activity is an important problem in computer vision that has been studied for over two decades [8], which traditionally involves predicting an activity class among a large set of common human activities (e.g., sports actions, daily activities). In contrast, here we are concerned with recognition of a limited set of gestures with a clear symbolic meaning (e.g., "follow me"). Recognizing and understanding gestures requires models that can represent the appearance and motion dynamics of the gestures. This can be accomplished by using (shallow) models that rely on hand-crafted features that capture detailed gesture knowledge, or more general deep models that are data-driven and less reliant on hand-crafted features [9]. Effectively, deep learning has been gaining increasing popularity in the domain of activity recognition [10] [11] [12] [13] [14] [15] [8] [16] [17]. This has been facilitated by the development of better learning algorithms and the exponential growth in available computational power (e.g., powerful GPUs). In this paper, we follow an approach based on a state-of-the-art deep learning model [18].

B. The Data Problem

Deep learning models, however, tend to have low sample efficiency and, thus, their success is contingent on the availability of large amounts of labeled training data. Collecting and annotating a large number of performances from human subjects is costly, time-consuming, error-prone, and comes with potential ethical issues. These problems are only likely to get exacerbated as society increases scrutiny on current data handling practices [19]. For these reasons, researchers have pointed that, rather than being limited by algorithm or computational power, in certain cases, the main bottleneck is the availability of labeled data [20]. To help mitigate this problem, several datasets are available for human activity recognition (for a full review see: [21] [8]) from diverse sources including movies [22], sporting events [23], or daily life [24]. However, a general problem is that they capture a very broad range of activities that is readily available on the web, or collected in heavily controlled environments. They are, therefore, unlikely to suffice for domains that target more specific types of activity and that favor models that are optimized for this subset of human activity, rather than general human activity. This is the case for our target domain

and, thus, new data must still be collected and annotated. To minimize the cost and problems associated with collecting labeled data from humans, we consider synthetic data.

C. Synthetic Data

Synthetic data may be critical to sustain the deep learning revolution. Synthetic data is less expensive, already comes with error-free ground truth annotation, and avoids many ethical issues [25]. Synthetic data, generated from simulations in virtual worlds, has been used in several important computer vision problems. In object and scene segmentation, synthetic data has been used to improve performance in object detection [26] [27] [28], object tracking [29] [30], viewpoint estimation [31], and scene understanding [29] [32] [33] [34]. In robot control domains, because of the difficulty and danger of training robots in real environments, synthetic data has often been used to teach robots visual space recognition [35] [36], navigation [37], object detection [38], and grasping control [39] [40] [41]. Synthetic virtual characters have also been used to train depth-based models for static pose estimation [42] [43] [44] [45], gaze estimation [46], and optical flow estimation [47]. Comparatively less work has looked at synthetic videos to recognize human activity, with Ionescu et al. [43] and de Souza et al. [10] being rare exceptions. Their work, though, focused on general activity recognition, rather than recognition of a specific set of gestures as we do here. Much of this research reports a performance bump due to augmenting real data with synthetic data (e.g., [10] [29] [44]) or competitive performance while training only on synthetic data (e.g., [31] [27]). Here we study whether synthetic data can also improve performance in our target domain with a focus on understanding how to optimize the value of synthetic data.

D. Approach & Contributions

We present a new synthetic dataset for a small set of control gestures (e.g., "follow me") generated from a custom simulator developed for our target domain. We also collected a small, by design, dataset of (real) data with human subjects. Since the focus of our study is on the value of synthetic data, rather than focus on model development, we simply use the state-of-the-art I3D model for activity recognition [18]. We present several experiments to optimize the parameters for this model when applying it to our domain, including number of training steps, real-to-synthetic data ratio, input resolution, input frames-per-second, and whether to use optical flow. Our results when training and testing exclusively on real data confirm that this is not an easy problem with an overall accuracy of only 52%. However, when augmenting the training data with synthetic data, our experiments show a 20% bump in performance.

To understand what was driving the contribution of synthetic data, we present ablation experiments where we held each parameter (skin color, background, gesture animations, etc.) of the synthetic data constant while varying the others; our results indicate a degradation varying from around 5% to 10%, thus suggesting that some parameters are more important than others. We test the performance of different synthetic data input resolution - namely, frame resolution, quality of rendering, and frames-per-second - on performance. Finally, we present a set of experiments testing whether the model is able to generalize to new gestures, when training exclusively on synthetic data (for that gesture); the results show robust performance for new gestures.

TABLE I
PARAMETERS USED TO GENERATE THE SYNTHETIC DATA.

Parameter	Range
Characters	Male civilian, male camouflage, female civilian, female camouflage
Skin color	Caucasian, African-American, East Indian
Thickness	Thin, thick
Animations	3 animations per gesture
Repetitions	Based on the distribution in the human data: Move in reverse, 2-4; halt, 1; attention, 1-3; advance, 1; follow me, 2-4; rally, 2-4; move forward, 2-4
Speeds	0.75×, 1.0×, 1.25×
Environments	Alpine, barren desert, coastal, rolling hills
Camera angles	0°, 45°, 90°, 135°, 180°, 225°. 270°, 315°

In sum, this paper makes the following contributions:

- A novel dataset of synthetic (and real) videos of control gestures that can be used as a benchmark for studying how models can be improved with synthetic data;
- A solution based on the I3D model for recognition of control gestures in human-robot teaming;
- Insight on the value of synthetic data for deep learning models.

II. THE DATA

For effective human-robot interaction, the gestures need to have clear meaning, be easy to interpret, and have intuitive shape and motion profiles. To accomplish this, we selected standard gestures from the US Army Field Manual [48], which describes efficient, effective, and tried-and-tested gestures that are appropriate for various types of operating environments. Specifically, we consider seven gestures (see Fig. 1A): *Move in reverse*, instructs the robot to move back in the opposite direction; *Halt*, stops the robot; *Attention*, instructs the robot to halt its current operation and pay attention to the human; *Advance*, instructs the robot to move towards its target position in the context of the ongoing mission; *Follow me*, instructs the robot to follow the human; and, *Move forward*, instructs the robot to move forward.

The human dataset consists of recordings for 14 subjects (4 females, 10 males). Subjects performed each gesture twice, once for each of eight camera orientations (0°, 45°, ..., 315°). Some gestures can only be performed with one repetition (halt, advance), whereas others can have multiple repetitions (e.g., move in reverse); in the latter case, we instructed subjects to perform the gestures with as many repetitions as it felt natural to them. The videos were recorded in open environments over four different sessions. The procedure for the data collection was approved by the US Army

Fig. 1. The gestures dataset: A, Reference gestures from the US Army Field Manual [48]; B, Real data examples; C, Synthetic data examples.

Research Laboratory IRB, and the subjects gave informed consent to share the data with the scientific community. The average length of each gesture performance varied from 2 to 5 seconds and 1,574 video segments of gestures were collected (see Fig. 1B for some examples). The video frames were manually annotated using custom tools we developed. The frames before and after the gesture performance were labelled 'Idle'. Notice that since the duration of the actual gesture - i.e., non-idle motion - varied per subject and gesture type, the dataset includes comparable, but not equal, number of frames for each gesture.

To synthesize the gestures, we built a virtual human simulator using a commercial game engine, namely Unity. The 3D models for the character bodies were retrieved from Mixamo[1], the 3D models for the face were generated on FaceGen[2], and the characters were assembled using 3ds Max[3]. The character bodies were already rigged and ready for animation. We created four characters representative of the domains we were interested in: male in civilian and camouflage uniforms, and female in civilian and camouflage uniforms. Each character can be changed to reflect a Caucasian, African-American, and East Indian skin color. The simulator also supports two different body shapes: thin and thick. The seven gestures were animated using standard skeleton-animation techniques. Three animations, using the human data as reference, were created for each gesture. The simulator supports performance of the gestures with an arbitrary number of repetitions and at arbitrary speeds. The characters were also endowed with subtle random motion for the body. The background environments were retrieved from the Ultimate PBR Terrain Collection available at the Unity Asset Store[4]. Finally, the simulator supports arbitrary camera orientations and lighting conditions.

The synthetic dataset was generated by systematically varying the aforementioned parameters as shown in Table I. In total, 117,504 videos were synthesized. The average video duration was between 3 to 5 seconds. To generate the dataset, we ran several instances of Unity, across multiple machines, over the course of two days. Fig. 1C shows examples of the synthetic data. The labels for these videos were automatically generated, without any need for manual annotation.

The full dataset, which we named Robot Control Gestures (RoCoG) Dataset, will be shared in an online repository and it will include all synthetic and real videos[5]. This dataset can be used to research and evaluate different models for gesture recognition, as well as the strengths and weaknesses of using synthetic data in human activity recognition.

III. The Model

Our model is based on the Inflated 3D convolutional (I3D) network [18], which has recently achieved state-of-art performance for activity classification. Since the focus of the paper is on studying the contribution of synthetic data rather than developing new models, using an existent state-of-the-art model suffices for our purposes. This model utilizes a pretrained image classification architecture by inflating the 2D convolutional filter and pooling kernels into a 3D (spatiotemporal) architecture. The Inception-v1 [49] model is used as the front-end 2D classification architecture, and pretrained sequentially on ImageNet and Kinetics[6]. Fig. 2 overviews the model architecture.

In this paper, we focused on the RGB stream rather than the two stream RGB + Flow I3D model. Traditional two-stream models for activity recognition leverage RGB as well as optical flow, and are generally designed for activity datasets recorded with a stationary camera. When the camera is in motion, such as being hand-held or mounted on a

[1]https://www.mixamo.com
[2]https://facegen.com
[3]https://www.autodesk.com/products/3ds-max
[4]https://assetstore.unity.com/packages/3d/environments/landscapes/ultimate-pbr-terrain-collection-72767
[5]The repository will be available at: https://vision.ece.ucsb.edu
[6]The Kinetics dataset is available at: https://deepmind.com/research/open-source/kinetics (last accessed on Aug-30, 2019)

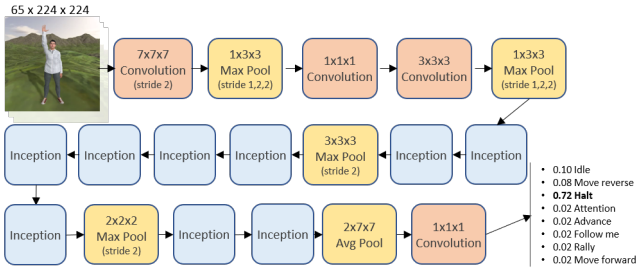

Fig. 2. Model overview.

robot, the flow channel will be dominated by ego-motion and is less informative to the activity [50]. While this can be compensated for, for simplicity we choose to simply remove this channel from the model. Moreover, adding the optical flow stream is not critical for our study of the value of synthetic data and would considerably slow down the preprocessing times and training times (given that we would have to split the available GPUs). Research also indicates that single RGB-stream 3D convolution networks are fully capable of learning motion patterns [18] [51]. Nevertheless, we still present one experiment comparing the best version of our RGB model with the two-stream model.

The resulting model consumes a 65-frame input video (i.e. $65\times224\times224$), and outputs a $N\times8$ classification corresponding to 8 subsequences of 8 frames each. N is the number of classes, which consists of the 7 gesture classes mentioned above, and an idle class that consists of background and transition motions between gestures. During training, we selected random 65 frame subsequences from the source video. The loss function was a conventional multiclass softmax cross entropy loss on the gesture class, with equal weights across all classes. The models were trained for up to 38k steps (see the Experiments section). We used exponential learning decay, with an initial learning rate set to 0.0001, a decay factor of 0.8, and with decay happening every 6,000 steps. When synthetic and real data were used to train a model, we weighted synthetic and real samples equally (see the Experiments section). All the code was implemented in Tensorflow. All models were trained on the same machine with an Intel 6-core CPU, 64 GB of RAM, and two NVIDIA RTX 2080 Ti GPUs.

IV. Experiments

Here we present a carefully selected subset of the experiments we conducted to achieve two goals: (1) optimize the parameters of the model, and (2) optimize the value of the synthetic data. In general, we tested against the *same* subset of the real data. Specifically, we used 10 subjects (7 males, 3 females) for training, and the remaining 4 subjects (3 males, 1 female) for testing. Notice that no human subject was used both for training and testing. To generate the train and test sets, the frames were extracted from the corresponding videos. In the rare occasions where there weren't enough frames for the 65-frame sliding window requirement, we padded the videos using the last frame. Recall that, since gesture performances can have different durations due to gesture type or individual differences, even though the test set contains an equal number of videos for each gesture class, in practice, it only had an approximately equal number of (non-idle) frames per gesture class.

Regarding synthetic data, we generated the training sets by sampling 10,000 (out of 117,504) videos. Limiting the synthetic sets to 10,000 videos considerably reduced the amount of data preprocessing times, frame set sizes, and training times, thus allowing us to run experiments more efficiently. Nevertheless, we made sure that the samples were balanced for gesture class (i.e., the sample had an equal number of videos for each gesture type), character model, and number of gesture animations per class. As before, padding was added at the end of the video, if necessary.

A. Baseline Experiments

To establish baseline performance, we trained and tested the model on real data, which revealed an accuracy of 51.5% (Fig. 3-A). We, then, trained exclusively on synthetic data and tested on the same real data test set, which showed an accuracy of 40.8% (Fig. 3-B). Notice that this is better than the accuracy for a model that makes random choices (12.5%) and a model that always classifies as 'Idle' (36.2%). The critical experiment, however, corresponds to training on synthetic and real data and testing on the real data test set, which led to an accuracy of 72.6% (Fig. 3-C) - i.e., a bump of about 20% over the first model. For completeness, we ran an experiment where training was done on real data and testing on synthetic data only; this led to 48.6% accuracy (not shown in the figure), suggesting that real data generalizes slightly better to synthetic data than the other way around.

B. Model Experiments

We ran experiments to optimize the number of training steps and the ratio of real to synthetic data. The performance accuracy for increasing amounts of training steps was: 17,976 steps, 65%; 35,952 steps, 72.6%; 71,904 steps, 71.1%; and, 107,856 steps, 67.1%. Thus, we found that 30k to 40k was a good range for the training steps. To study the impact of real to synthetic ratio in the training data, we created different training sets by repeating the number of real videos according to the ratio (e.g., we repeated the real videos about $6\times$ for the 1:1 ratio). The performance for different ratios was: 1:1, 72.6%; 2:1, 60.4%; 3:1, 59.3%; and, 4:1, 50.8%. Therefore, a 1:1 ratio led to the best performance. We also wanted to test whether a dual-stream model, which includes optical flow, would lead to a performance gain. The results were: RGB Only, 72.6%; Optical Flow Only, 54.5%; and, RGB + Optical Flow, 73.6%. Thus, optical flow did lead to a minor improvement; however, as argued in Section III, for efficiency, we conducted the remaining tests using the RGB-only model as a reference.

C. Input Resolution Experiments

We ran three experiments to understand the impact of synthetic data frame pixel resolution, rendering quality, and

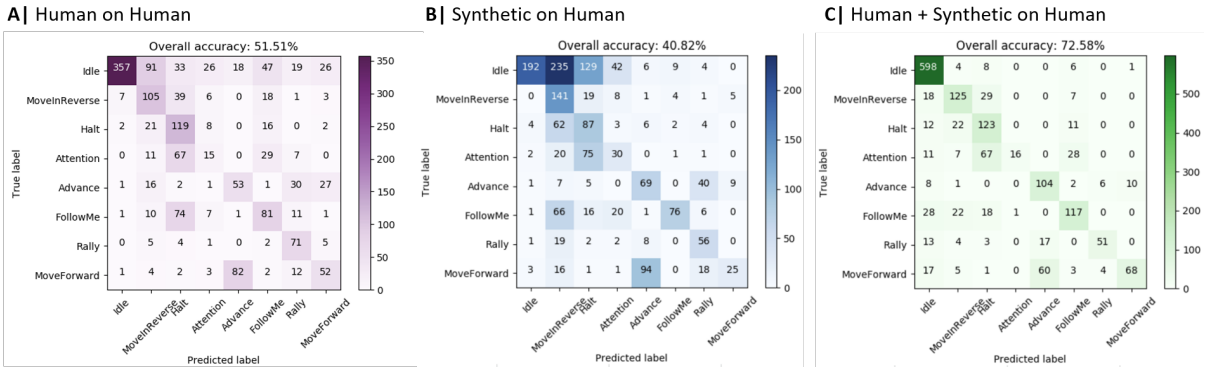

Fig. 3. Gesture recognition accuracy and confusion matrices for the baseline experiments: (A) Trained and tested on human data; (B) Trained on synthetic data, tested on human data; (C) Trained on human and synthetic data, tested on human data.

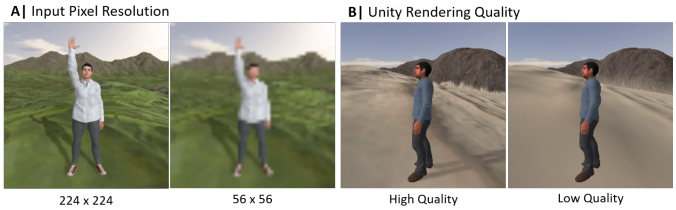

Fig. 4. Examples for the input pixel resolution (A) and Unity rendering quality (B) experiments.

frames-per-second resolution. Regarding pixel resolution, we compared the original 224 $\times$ 224 resolution to reduced resolutions. This was done by downsampling the original synthetic frames to the target resolution (e.g., 56 $\times$ 56) and then upsampling them back to 224 $\times$ 224 (Fig.4-A). We then trained the model on the modified synthetic data and tested on the real data test set. The critical comparison was, thus, with respect to the baseline model trained on synthetic data and tested on real data (Fig. 3-B). The results were: 224 x 224 (reference), 40.8%; 56 x 56, 24.8%; 112 x 112, 43.8%; and, 168 x 168, 37.1%. Thus, there was little to no degradation for 168 $\times$ 168 and 112 $\times$ 112 but, a significant reduction in performance for 56 $\times$ 56. Secondly, we generated a low resolution version of the synthetic data using the lowest rendering quality for Unity (no shadows, reduced texture quality, no anti-aliasing, etc.), and compared to the baseline version of the synthetic data generated with the highest rendering quality (Fig. 4-B). The results showed that the low resolution version had lower performance (35.8%) when compared to the baseline (40.8%). Finally, we manipulated the frames-per-second (fps) resolution of the input video. The original video was sampled at 30 fps and we compared to the performance when the input video was at 15 fps and 10 fps. This was accomplished programmatically by adjusting which frames were fed through the input queue in real-time (e.g., for 15 fps, the code would just skip every other frame). The test set's fps was also adjusted accordingly. The results were: 30 fps (reference), 40.8%; 15 fps, 36.7%; and, 10 fps, 30.1%. Therefore, reducing the fps led to an approximately linear reduction in performance.

TABLE II

GESTURE RECOGNITION ACCURACY FOR THE ABLATION EXPERIMENTS.

Experiment	Accuracy (%)
Synthetic on Human (reference)	40.8
Female characters only	37.3
One character only (Male civilian)	29.7
Caucasian skin color only	26.2
0.75x speed only	28.0
Coastal terrain only	41.9
Thin character only	28.8
One gesture animation only	32.1

D. Ablation Experiments

To get insight on the relative contribution of each synthetic data parameter, we ran experiments using ablated versions of the synthetic dataset. The method consisted of keeping all parameters unchanged except for the targeted parameter, the range of which was constrained. We ran ablation tests for single character, only female characters, single skin color, single thickness level, only one animation per gesture class, single speed, and always the same background. These ablated datasets were then used to train models and tested on the real data test set. The critical comparison was against the model trained on synthetic data and tested on real data. The results are shown in Table II. The experiments confirmed a significant reduction in performance of about 10% for all domain parameters, except for gender and terrain type which did not lead to meaningful degradation in performance.

E. Generalization Experiments

Since we do not focus on general activity recognition, it should be unsurprising that our dataset has a much smaller number of target classes than some of the existing activity recognition datasets - for instance, Chalearn [7] and Kinetics have hundreds of target classes. However, as indicated by the baseline experiments results, the current challenge is not easy and there is still much room for future improvement. An important question, though, is: Can we use synthetic data to generalize to new gestures that are not in the original set? This has practical importance as it speaks to generalization of

TABLE III

GESTURE RECOGNITION ACCURACY FOR GENERALIZATION EXPERIMENTS. THE 'BASELINE ACCURACY' CORRESPONDS TO RECOGNITION RATE FOR TARGET GESTURES IN THE HUMAN + SYNTHETIC BASELINE EXPERIMENT. THE 'GENERALIZATION ACCURACY' CORRESPONDS TO RECOGNITION FOR THE TARGET GESTURE WHEN TRAINING THAT GESTURE ONLY ON SYNTHETIC DATA.

Gesture	Baseline (%)	Generalization (%)
Advance	74.81	79.39
Attention	33.33	12.40
Follow Me	48.92	62.90
Move Forward	73.42	43.04
Halt	86.31	73.21
Rally	72.73	57.95
Move in Reverse	62.57	69.83

the model. To test this, we ran experiments where we trained the model on real and synthetic data for all gestures, except one which was only trained on synthetic data. Presumably, in practice, it is easier to synthesize data for a new gesture than to collect real data for it. We ran seven experiments, one for each gesture in our set. The critical comparison was to the baseline per-gesture accuracy, when training on the real and synthetic datasets. Table III shows the results. In some cases the performance was lower (e.g., halt), whereas in others it was higher (e.g., advance); however, overall, the majority of the gestures had relatively high accuracy ($> 55\%$), even though training occurred only on synthetic data.

V. DISCUSSION

In this paper, we present insight, that may also inform other domains, on how to optimize the generation and impact of synthetic data for gesture recognition. First, our results suggest that increasing photorealism leads to improved performance. The experiment on frame pixel resolution shows that performance held relatively well when the input resolution was moderately decreased (from 224×224 to 168×168 or 112×112), but was noticeably reduced with considerable visual degradation (56×56). Another experiment indicated that rendering the synthetic data with low quality led to a clear reduction in performance. Second, our results indicate that, at least in this domain, the *motion* realism of the synthesized data is particularly important. Effectively, when reducing the frames-per-second resolution for the input video - thus, reducing its motion fidelity - the performance showed proportional degradation. Future work should further explore the relative value of photo and motion realism for performance and compare the present dataset with higher quality versions (e.g., motion-captured animation).

The results confirm that it is critical to synthesize data that is fully representative of the target domain. Our results revealed that most of the parameters in our simulator had a measurable impact on the performance of the synthesized data, including the number, skin color, and thickness of the characters, and the speed and number of gesture animations. On the other hand, as may have been expected, supporting many background types had minimal impact on gesture recognition performance. We note that, from a practical point of view, this experimental approach of systematically testing each domain parameter can be useful for optimizing development efforts for the synthetic data simulator.

The present work used the I3D model as is, but future work should consider domain adaptation or style transfer techniques to improve performance. Domain adaptation techniques encourage the model to learn domain-invariant representations - in our case, that wouldn't distinguish between synthetic and real data - and they have shown to improve the performance of models trained on simulated data [52] [39] [53] [54] [46] [55] [56]. Style transfer techniques, on the other hand, focus on changing the (pixels in the) data in one domain (e.g., synthetic) to better match another domain (e.g., real) [57] [58] [59]. Furthermore, there has been promising research in new activity recognition models including models that account for optical flow while adjusting for camera ego motion [60] [61], and models that use non-local operations [62].

VI. CONCLUSION

Gesture communication is important for successful human-robot collaboration in vast, dynamic, outdoor environments. Here we share a dataset of real and synthetic control gestures that is useful for training solutions for this problem, propose a first solution based on the state-of-the-art I3D model and, more importantly, demonstrate that synthetic data can be crucial in overcoming the challenge of collecting vast amounts of annotated human subject data. As society increasingly scrutinizes the use of (real) data and as technology makes it easier to develop custom simulators, synthetic data is likely to play an increasingly pervasive role in the deep learning revolution and experimentation, such as presented here, becomes essential to optimize the value of synthetic data.

VII. ACKNOWLEDGMENTS

This research was supported by the US Army. The content does not necessarily reflect the position or the policy of any Government, and no official endorsement should be inferred.

REFERENCES

[1] T. Fong and I. Nourbakhsh, "Interaction challenges in human-robot space exploration," *NASA Technical Reports Server, volume =*.

[2] A. Kott and D. Alberts, "How do you command an army of intelligent things?" *Computer*, vol. 50, no. 12, pp. 96–100, 2017.

[3] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry: Intuitive human-robot communication from human observation," in *Proc. of HRI'13*, 2013.

[4] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[5] D. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics & Applications*, vol. 14, no. 1, pp. 30–39, 1994.

[6] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and et al., "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proc. of CVPR'16*, 2016.

[7] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and et al., "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition," in *Proc. of CVPR Workshops'16*, 2016.

[8] M. Vrigkas, C. Nikou, and I. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, pp. 1–28, 2015.

[9] C. de Souza, A. Gaidon, E. Vig, and A. López, "Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition," in *Proc. of ECCV'16*, 2016.

[10] C. de Souza, A. Gaidon, Y. Cabon, and A. López, "Procedural generation of videos to train deep action recognition networks," in *Proc. of CVPR'17*, 2017.

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of CVPR'16*, 2016.

[12] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proc. of CVPR'14*, 2014.

[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. of NIPS'14*, 2014.

[14] O. Ulutan, A. Iftekhar, and B. Manjunath, "Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions," in *Proc. of IEEE/CVF'20*, 2020.

[15] O. Ulutan, S. Rallapalli, M. Srivatsa, C. Torres, and B. Manjunath, "Actor conditioned attention maps for video action detection," in *Proc. of IEEE Winter Conference on Applications of Computer Vision'20*, 2020.

[16] X. Wang, A. Farhadi, and A. Gupta, "Actions transformations," in *Proc. of CVPR'15*, 2015.

[17] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. of CVPR'15*, 2015.

[18] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. of CVPR'17*, 2017.

[19] P. Leskin and N. Bastone, "The 18 biggest tech scandals of 2018," https://www.businessinsider.com/biggest-tech-scandals-2018-11, 2018.

[20] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. of ICCV'17*, 2017.

[21] F. Heilbron, V. Escorcia, B. Ghanem, and J. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. of CVPR'15*, 2015.

[22] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. of CVPR'08*, 2018.

[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. of CVPR'14*, 2014.

[24] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proc. of CVPR'14*.

[25] A. Gaidon, A. Lopez, and F. Perronnin, "The reasonable effectiveness of synthetic visual data," *International Journal of Computer Vision*, vol. 126, pp. 899–901, 2018.

[26] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 961–972, 2018.

[27] A. Prakash, S. Boonchoon, M. Brophy, D. Acuna, E. Cameracci, and at al., "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *Proc. of ICRA'19*, 2019.

[28] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, and et al., "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. of the IEEE CVPRW'18*, 2018.

[29] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. of CVPR'16*, 2016.

[30] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications?" *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[31] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?" in *Proc. of the Computer Vision-ECCV Workshops'16*, 2016.

[32] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. of CVPR'16*, 2014.

[33] S. Richter, V. Vineet, S. Roth, and K. Vladlen, "Playing for data: Ground truth from computer games," in *Proc. of ECCV'16*, 2016.

[34] A. Shafaei, J. Little, and M. Schmidt, "Play and learn: Using video games to train computer vision models," in *Proc. of BMVC'16*, 2016.

[35] E. Sizikoval, V. Singh, B. Georgescu, M. Halber, K. Ma, and T. Chen, "Enhancing place recognition using joint intensity - depth analysis and synthetic data," in *Proc. of ECCV, VARVAI Workshop'16*, 2016.

[36] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, and et al., "Embodied question answering in photorealistic environments with point cloud perception," in *Proc. of CVPR'19*, 2019.

[37] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," in *Proc. of RSS'17*, 2017.

[38] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *Proc. of ICRA'12*, 2012.

[39] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, and et al., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *Proc. of ICRA'18*, 2018.

[40] A. Saxena, J. Driemeyer, and A. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[41] R. 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. of IROS'17*, 2017.

[42] H. Hattori, N. Lee, V. Boddeti, N. Feainy, and K. K. et al., "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, pp. 1027–1044, 2018.

[43] C. Ionescu, P. Dragos, O. Vlad, and S. Cristian, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[44] A. Shafaei and J. Little, "Real-time human motion capture with multiple depth cameras," in *Proc. of CRV'16*, 2016.

[45] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, and et al., "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2013.

[46] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. of CVPR'17*, 2017.

[47] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, and C. H. et al., "Flownet: Learning optical flow with convolutional networks," in *Proc. of ICCV'15*, 2015.

[48] U. Army, "Us army field manual 21-60: Visual signals," 1987.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, and et al., "Going deeper with convolutions," in *Proc. of CVPR'15*, 2015.

[50] S. Siravan and M. Trivedi, "A review of recent developments in vision-based vehicle detection," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2013.

[51] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. of ECCV'10*, 2010.

[52] C. Atapattu and B. Rekabdar, "Improving the realism of synthetic images through a combination of adversarial and perceptual losses," in *Proc. of IJCNN'19*, 2019.

[53] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, and H. L. et al., "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, pp. 1–35, 2016.

[54] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. of ICML'17*, 2018.

[55] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. of AAAI'16*, 2016.

[56] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. of CVPR'17*, 2017.

[57] L. Gatys, A. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. of CVPR'16*, 2016.

[58] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. of ECCV'16*, 2016.

[59] F. Luan, S. Paris, E. Schechtman, and K. Bala, "Deep photo style transfer," in *Proc. of CVPR'17*, 2017.

[60] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. of CVPR'15*, 2015.

[61] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. of CVPR'18*, 2018.

[62] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. of CVPR'19*, 2019.