Learned Hand Gesture Classification through Synthetically Generated Training Samples

Kyle Lindgren[1], Niveditha Kalavakonda[1], David E. Caballero[1], Kevin Huang[2], Blake Hannaford[1]

Abstract— **Hand gestures are a natural component of human-human communication. Simple hand gestures are intuitive and can exhibit great lexical variety. It stands to reason that such a user input mechanism can have many benefits, including seamless interaction, intuitive control and robustness to physical constraints and ambient electrical, light and sound interference. However, while semantic and logical information encoded via hand gestures is readily decoded by humans, leveraging this communication channel in human-machine interfaces remains a challenge. Recent data-driven deep learning approaches are promising towards uncovering abstract and complex relationships that manual and direct rule-based classification schemes fail to discover. Such an approach is amenable towards hand gesture recognition, but requires myriad data which can be collected physically via user experiments. This process, however, is onerous and tedious. A streamlined approach with less overhead is sought. To that end, this work presents a novel method of synthetic hand gesture dataset generation that leverages modern gaming engines. Furthermore, preliminary results indicate that the dataset, despite being synthetic and requiring no physical data collection, is both accurate and rich enough to train a real-world hand gesture classifier that operates in real-time.**

I. Introduction

Hand gestures provide a particularly intuitive interface, and may complement more commonplace human-machine interaction modalities (e.g keyboard, touchscreen) thereby enabling a repertoire of personalized user inputs from familiar hand motions. Crucially, hand gestures exhibit compatibility with existing human-to-human gestures and lack physical contact constraints, all while being robust to radio emissions or sound and light pollution. However, reliably and generally recognizing hand gestures is a difficult task; obtaining sufficiently descriptive training data is practically challenging and not well understood. Collection of real-world training samples requires onerous human effort and often fails to adequately characterize novel environments and gestures. Additionally, empirical evidence concerning training data variability and recognition performance with respect to supervised learning lacks rich optimization knowledge, thus exacerbating the dataset generation issue.

Most recent advances in gesture recognition involve deep learning, as model-free approaches have proven more adept than model-based for solving highly complex tasks. One of the major challenges in this technique, however, is that it requires vast amounts of data for training. This training data must contain content rich enough to characterize anticipated variability in real-world implementations. In contrast with other domains for which large datasets are readily available, such as computer vision, speech recognition, search engines, DNA sequencing, and stock market analysis [1] [2], training models for recognizing gestures is especially difficult due to the lack of readily available training data. One way to alleviate this problem is by the process of data augmentation, which involves synthetically generating dense datasets from currently available sparse datasets, leading to more effective training [3].

This paper presents a refined data generation technique which leverages modern game engines to produce rich and realistic, purely synthetic and perfectly labeled training data of human hand gestures. This method affords efficient dataset generation without the need of human subjects or physical data collection, and its effectiveness is demonstrated with the training of a real-time hand gesture classifier. The entire process consists of two major components, as shown below in Fig 1, and is described in detail in Section III.

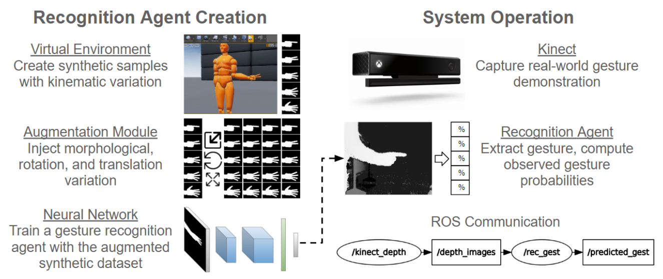
Recognition Agent Creation
System Operation
Virtual Environment
Create synthetic samples with kinematic variation
Kinect
Capture real-world gesture demonstration
Augmentation Module
Inject morphological, rotation, and translation variation
Recognition Agent
Extract gesture, compute observed gesture probabilities
Neural Network
Train a gesture recognition agent with the augmented synthetic dataset
ROS Communication
/kinect_depth
/depth_images
/rec_gest
/predicted_gest

Fig. 1. Overall system workflow consists of two separate processes: (left) creating gesture recognition agent (right) implementing real-time classification of real-world gestures

A. Contribution

To the best of the authors' knowledge, this work is the first to

1) utilize modern game engines to synthetically generate hand gesture datasets for classifier training.
2) demonstrate gesture classification from purely synthetic data in real-time.

Preliminary results are promising to the use of synthetically generated geometric datasets for real-time static gesture recognition. The method is extendable to other surface geometry based classification tasks.

[1]Kyle Lindgren, Niveditha Kalavakonda, David E. Caballero and Blake Hannaford are with the University of Washington, Dept. of Electrical Engineering, 185 Stevens Way, Paul Allen Center - Room AE100R, Campus Box 352500, Seattle, WA 98195-2500, USA. {kyle509, nkalavak, dcaballe, blake}@uw.edu

[2]Kevin Huang is with Trinity College, Dept. of Engineering, 300 Summit St, Hartford, CT 06106 USA kevin.huang@trincoll.edu

II. Background

A. Related Work

Computer vision approaches are preferred solutions to gesture recognition due to their ability to track and recognize static and dynamic gestures without depending on cumbersome physical hardware to be worn by the user. There are two broad categories of computer vision approaches: marker-based gesture recognition and markerless gesture recognition [4]. Due to minimal hardware requirements and non-intrusive nature, markerless systems are more widely researched and implemented. Various approaches involving markerless recognition have been introduced over the last decade. These include hand-crafted features such as skin-color detection [5], shape models and particle filtering [6]. Other groups have also approached gesture tracking with distinctly colored gloves [7] or a controlled background [8].

Gesture recognition using hand-crafted features have been successful in niche cases, generally for static gestures. Bonansea et al. [9] utilized ZCam and Support Vector Machines (SVM) to recognize one-hand gestures. Wang et al. [10] and Chen et al. [11] used statistical feature-based classifiers implementing hidden Markov models to classify hand gestures. Other implemented pattern classifiers include principal component analysis [12], a combination of histogram of gradient features and SVM [13] [14] and k-nearest neighbors (k-NN) [15]. However, because of the variability in the types of acquisition devices, skin color, illumination, background and minute intrapersonal differences in gesture execution, hand-crafted feature classification methods require additional training for each captured image frame of a gesture. This approach can thus be quite inefficient for real-time implementation.

Other methods are better suited for real-time performance. Marcel et al. used a constrained generative model to fit a probability distribution of a set of hands by incorporating a non-linear compression neural network and examples of non-hand images [16]. Molchanov et al. employed spatio-temporal data augmentation to the VIVA challenge dataset to avoid overfitting [17]. The authors made use of online (affine transformations, spatial elastic deformation, fixed-pattern drop-out and random drop-out) and offline (reversing order and/or mirroring) data augmentation. Two sub-networks, a high-resolution network and a low-resolution network, were developed to evaluate the performance of convolutional neural network classifiers on spatio-temporal volumes generated by interleaving depth and intensity channels. The peak accuracy of this method was further shown to be 74.4%, an improvement from methods relying on hand-crafted features only.

Additionally, Tsai et al. explored the effects of using synthetically generated samples to train a neural network hand classifier [18]. They concluded that adding a relatively small amount of real-world samples to their training dataset drastically improved recognition performance, and attributed this to the fact that the color of their synthetic samples greatly deviated from those of the real-world. A 37.5% to 77.08% jump in recognition performance implies dependence of their methods on inclusion of real-world samples.

These works do not address training data efficiency, and supporting new gestures requires extensive human effort.

III. Experimental Methods

A. Gesture Recognition Agent

1) Virtual Gesture Samples: Virtual training dataset creation starts with the Unreal Engine 4 game engine. Hand kinematics of a virtual avatar are manipulated to display hand gestures, as shown in Figure 2, while depth imagery is captured using Microsoft's AirSim plug-in [19]. Depth imagery was utilized since depth samples of synthetic environments correlate well with real-world counterparts, fitting domain adaptation well, whereas greater levels of realism with RGB imagery exhibits large discrepancies [20].

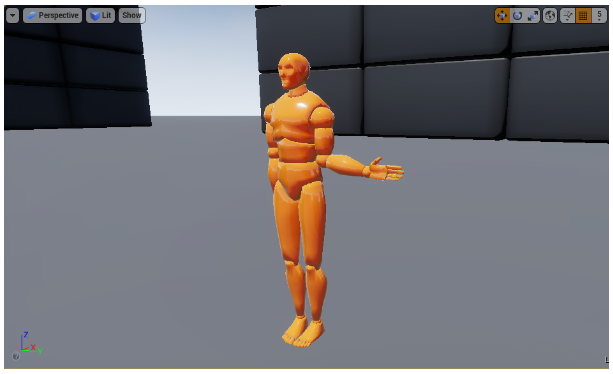

Fig. 2. Virtual avatar displaying representative hand gesture, rendered in Unreal Engine 4.

A total of five hand gestures were tested, and are labeled as 1, 2, 3, 4, 5. These names indicate the total number of outstretched digits within the hand gesture. The task is to classify real-world data and hand gestures using a classifier trained solely on synthetic data. Figure 3 depicts both virtual and real-world samples of the five tested gestures.

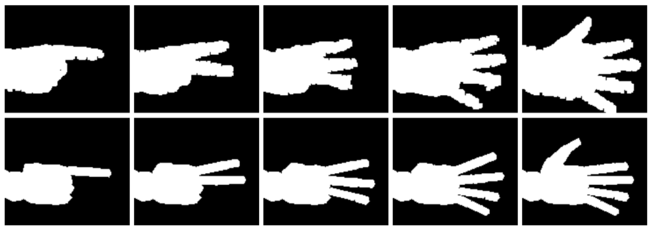

Fig. 3. Samples of all five gestures in the real-world (top) and virtual (bottom) environments. These samples are taken after the extraction and background removal phase, which is explained in Section III-B.2

2) Dataset Augmentation: Variability was injected into the captured dataset to enhance breadth of the training data. Captured depth samples of the virtual gestures undergo several levels of augmentation including scale (morph), rotation, and translation. These modifications are applied systematically and consistently across seed synthetic gestures. The entire image augmentation pipeline is depicted in Figure 4.

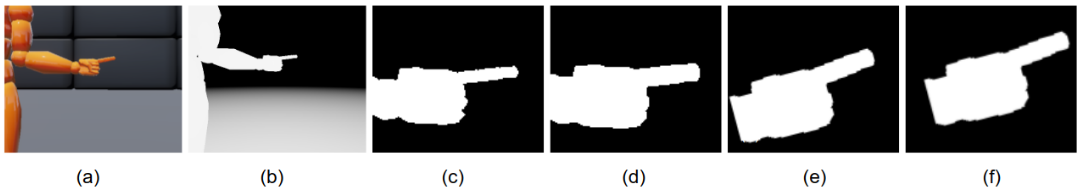

Fig. 4. From left to right: (a) scene in the virtual environment, (b) raw depth capture, (c) preprocessing step to remove background and extract hand (template matching), (d) morphological variation, (e) rotational variation, (f) translational variation.

Additionally, prior to the synthetic data augmentation, the virtual avatar finger angles are systematically varied in software. This introduces kinematic variations prior to depth capture by AirSim. This kinematic variation alters finger angles with respect to the palm. Constraining factors prevented unrealistic alterations. In particular, modifications considered finger overlap in the depth images and limited the range of angles to 20 degrees. For each gesture, samples were collected for all configurations with fingers being in one of three angles (20 degree range, in 10 degree increments). Samples with excessive overlap were discarded.

Morphological operations enhance size robustness to samples by expanding or shrinking depth silhouettes. Four separate morphological operations were performed, one erosion and three dilations. Operations are skewed towards dilations to better represent the Kinect's observed tendency to overestimate width of relatively fine objects such as fingers. All morphological kernels were square in shape, with their edge lengths being 2, 2, 4, and 6 pixels, respectively.

Additional augmented data were produced through sample rotation. Specifically, new samples were generated from each original sample via six rotations about the optical axis in increments of 5 degrees (generated samples were rotated versions, -15 to 15 degrees, of the original). Rotation augmentations were only considered about one degree of freedom in an effort to reduce experiment complexity, and since performed morphological variations can account for a range of rotations about other axes.

Virtual depth samples were also translated within their images at four levels of alteration. These included translations with magnitude of 20 pixels in the directions of 45, 135, 225 and 315 degrees.

TABLE I
DATASET SIZES

Gesture	K	KMRT	*T	*R	*M
1	3	525	105	75	105
2	8	1400	280	200	280
3	9	1575	315	225	315
4	9	1575	315	225	315
5	15	2625	525	375	525
Total	44	7700	1540	1100	1540

Table I summarizes the dataset sizes across gestures and variations. Variations are abbreviated with K–kinematic, M–morphological, R–rotation and T–translation. * denotes exclusion of one variation (e.g. rotation $\notin$ *R).

3) Network Architecture: Tensorflow served as the backend for neural network creation and training. Inputs to the network were grayscale images (64x32), while classification probability measures for each gesture were generated outputs. The overall network structure is illustrated in Figure 5, and considered several techniques to avoid overfitting to the training dataset. This was crucial since training data was purely synthetic, while the learnt behavior was used to classify in a new domain, real-world gestures. These overfitting avoidance strategies included limiting the depth of the network, adding batch normalization between layers, and setting dropout parameters relatively high (0.5) between layers. Rectified linear activation layers were used for all layers except the last where softmax was used to format the output as class-conditional gesture probabilities.

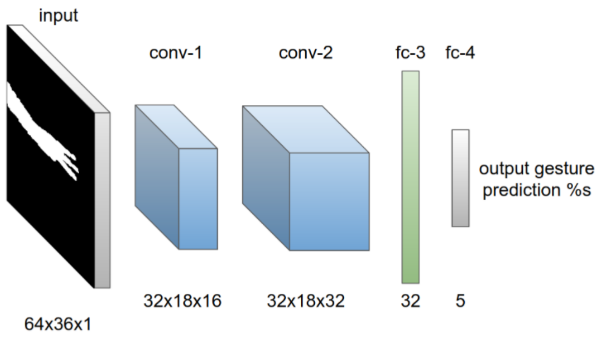

Fig. 5. Extracted and downsized hand gesture depth images underwent two convolutional layers and two fully connected layers, with the last layer serving as a softmax classifier.

4) Training: Four distinct gesture recognition agents were trained using the network structure described in Figure 5. Each classifier used training datasets across several levels of sample variation, as shown in Table I. Some noteworthy specifications about the implemented training process are:

- Training was performed for only 5 epochs with 20% of the samples comprising the validation set.
- Individual gesture sets were increased in size to match that of the largest gesture set (gesture 5 in all cases) by randomly duplicating samples. This step reduced potential for gesture bias.
- The number of episodes for each network served as a multiplier for all sets to contain the same number of samples. As such, *T and *M included 5 episodes while *R included 7, resulting in 7700 samples per classifier training set.

- Training was performed with an NVIDIA GeForce GT 640M LE GPU, and never exceeded 2 minutes.

B. System Operation

Primary system components included the Kinect v2 RGB-Depth camera and the gesture recognition module. These components utilized ROS to support communication across multiple platforms, as depicted in Figure 6.

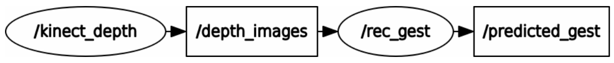

Fig. 6. Basic ROS architecture: Kinect publishes raw depth images topic, to which the gesture recognition module subscribes. The recognition module then publishes the predicted gesture.

1) Depth Camera: The Kinect v2 time of flight RGB-Depth camera was the image sensor used in this work. It captures grayscale depth images at 30 fps with a 512x424 resolution, depth range of 0.5m to $\sim$8m, and a 70x60 degree FOV. Figure 7 shows a typical raw depth image captured by the Kinect v2 of gesture 1.

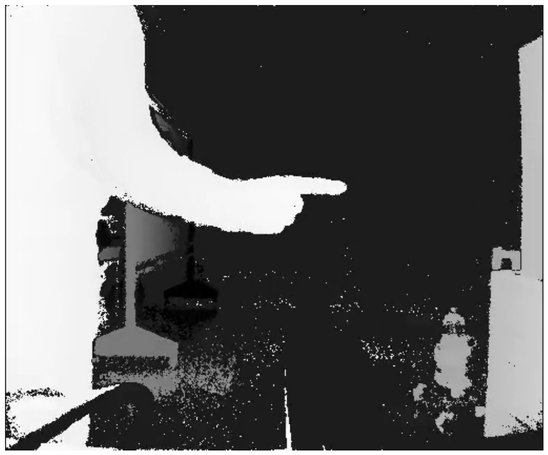

Fig. 7. Typical raw depth image of gesture 1 captured by Kinect v2.

2) Gesture Recognition: Raw depth images from the Kinect v2 camera were preprocessed in several steps prior to gesture recognition agent classification or training:

1) Template matching – a template matching procedure as depicted in Figure 4(c), was used to isolate and extract the hand from the rest of the scene, resulting in 142x122 images.
2) Depth threshold filtering – depth readings beyond a static threshold distance of the extracted hand were ignored to remove background clutter.
3) Morphological opening – a morphological opening operation with a square 5x5 kernel reduced specular noise.
4) Compression – the resulting images were then downsized to 64x32 with area-averaging interpolation in order to reduce moiré patterns.

The third step in the image preprocessing aided in ameliorating noise issues exhibited by the Kinect v2. Following this step, the preprocessed images formed the input for the gesture recognition agents. The gesture predictions were published as an array embedded in a ROS topic. Average latency between image frames captured from the Kinect v2 sensor and gesture predictions output by the recognition module were measured at approximately 15ms.

C. Experiment Design

Several data augmentation dimensions (morphology/scale, rotation and translation) were implemented on the synthetic dataset and in various combinations. These combinations formed test conditions for the four gesture recognition agents, and are named KMRT, *T, *R and *M. The KMRT dataset contains all forms of data augmentation while the * datasets contain all but one form, as identified in their name. Dataset sizes are shown in Table I.

Classifiers trained with the various augmentations were implemented in real-time (15 ms latency) and evaluated with real world gestures. Two quantitative measures were of interest in this experiment:

1) gesture angle — range of tolerable gesture rotation deviation for robust classification
2) gesture distance — range of tolerable gesture distance deviation for robust classification

Gesture recognizing networks were separately trained using the same network architecture. Furthermore, these networks were trained with datasets consisting of purely virtual hand gestures generated via the Unreal 4 engine. Again, these synthetic data were systematically augmented in varying combinations of augmentation dimensions. Gesture recognition robustness with real-world data was quantified by the measured limits of deviation in rotation and distance at which recognition proved accurate — incremental deviations from center of workspace were performed until the first failure was observed for that trial. Three different users with noticeably different hand morphologies participated in these experiments. Each subject underwent the same data collection protocol. Specifically, for each gesture type (1-5) they were instructed to:

1) stand 60 cm away and facing the Kinect v2 sensor with left hand horizontal (fingers parallel to the ground), ensure hand is within FOV of the sensor.
2) rotate the gesture making hand clockwise, then counter-clockwise until failed classification.
3) while varying finger joint angles, translate the gesture making hand from the center of the workspace towards the sensor until failed classification – repeat for moving away from the sensor.

Rotation measurements were limited at 60 degrees in either direction, the recorded maximum tolerance in rotation. Distance measurements were constrained to a minimum distance per Kinect v2 limitations (50cm) and at a maximum by image preprocessing methods for background noise removal (69cm). Again, classification failure was determined at the first instance within the tested variation dimension (rotational or translational) for which classification was incorrect. Four networks were tested with each of the three subjects.

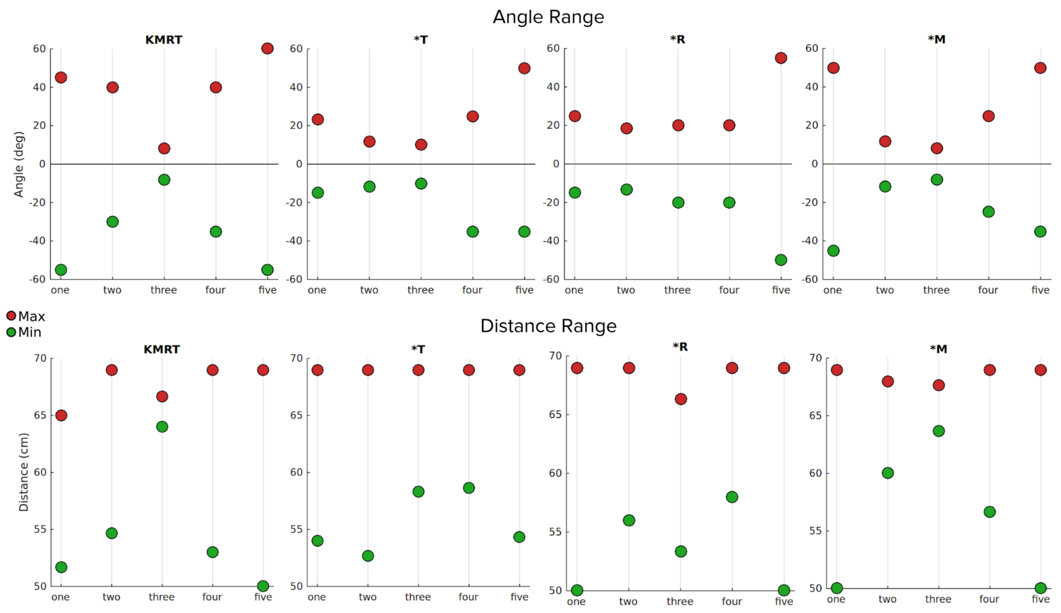

Fig. 8. Classifier robustness to operable angle (top) and distance (bottom) range. Data points are the mean reported failure limits of rotation and translated distance amongst the three subjects.

The four networks, KMRT, *T, *R and *M, were trained using synthetic datasets with different combinations of augmentation. Results from this user study help to assess the relative impact of each of these augmentation dimensions with regard to building robust learned behavior from synthetic hand gesture training data for real-world classification.

IV. Results

Two main quantitative metrics were evaluated. Classifier performance with respect to hand gesture angle and distance range are summarized in Figure 8 and Tables II and III.

TABLE II
Gesture Average Ranges

Gesture		KMRT	*T	*R	*M
1	ang	100	38	40	95
	dist	13	15	19	19
2	ang	70	24	31	24
	dist	14	16	13	8
3	ang	16	20	40	16
	dist	3	11	13	4
4	ang	75	60	40	50
	dist	13	10	11	12
5	ang	115	85	105	85
	dist	19	15	19	19

For rotation, subjects rotated the hand gesture clockwise at the center of the workspace until the first failed classification was observed. This was then repeated for counter-clockwise rotation. For distance, subjects translated the hand gesture towards the Kinect v2 sensor from the center of the workspace until the first failed classification was observed. This was then repeated for translation away from the sensor.

V. Discussion

In terms of robustness to rotational variation, the KMRT classifier performed the best overall, displaying particularly good performance with gesture ‘2’. All networks struggled with gesture ‘3’. It is also worth noting that the *R classifier, the network lacking rotational variation within the training dataset, does not show significantly poorer rotational robustness compared to that of the other networks. For robustness to translational distance variation, gesture ‘3’ proved again to be most difficult to classify. Interestingly, the *T and *R networks exhibit improved performance over the KMRT.

TABLE III
Summary of Results: Average Range for all Gestures

Measure		KMRT	*T	*R	*M
Angle	Range (deg)	75.2	45.4	51.2	54.0
	Effect (%)	-	-39.6	-32.0	-28.2
Distance	Range (cm)	13.0	13.4	15.0	12.4
	Effect (%)	-	+3.1	+15.4	-4.6

Generally, gestures 2, 3, and 4 were more difficult to classify, as illustrated by Figure 8. This is a somewhat anticipated result due to their geometric similarity with one another. Additionally, kinematic and morphological dilations during sample augmentation occasionally resulted in samples with non-distinct fingers. Such samples were left in the datasets in an effort to replicate observed noise exhibited by the Kinect v2 sensor. Images captures by the Kinect v2 was also prone to exhibit non-distinct fingers, generally caused by uniform expansion of thin/fine objects. Unfortunately, the recognition agent was unable to robustly characterize combined fingers as two or more separate fingers physically grouped together. Further analysis on the effects of size/digit thickness variation on the synthetic dataset may provide a solution to alleviate the combined finger issue.

Translational variation was included in the sample augmentation pipeline to help account for real-world sample variance in gesture location within the image. Such variations can occur from potential noise or alignment differences during template matching in gesture extraction. A similar template matching technique was used with the virtual environment samples to extract gestures, but locating the palm (matched template) in the noise-free environment performed more consistently as compared to the same real-world task. As the results demonstrate in Table III, inclusion of translational variation in the synthetic training set aided in gesture recognition. Increasing translational variation may lead to improved performance with template matching and other imperfect object detection methodologies.

Furthermore, Table III illustrates the importance of translational variation in the synthetic data generation. Morphological and rotation variation may also assist in improving angle robustness for gesture classification. As expected, the lack of morphological variation slightly degraded recognition of gestures at different ranges. Surprisingly, results suggest that rotation variation may inhibit robustness to varying distances. This can be attributed to the fact that subjects used the canonical (non-rotated) hand gesture during translational distance testing procedures.

Noticeable morphological differences between virtual and real-world gestures, as demonstrated in Figure 3, illustrate the utility of the proposed methods. Hand gesture recognition was achieved in this work despite lacking accurate modeling of hand kinematics and zero physical dataset collection. The neural networks trained with entirely synthetic data were able to classify hand gestures demonstrated by real human users exhibiting different hand characteristics and morphologies. Stark inter-subject contrast in palm and finger shape, as well as variation in placement of non-displayed fingers are a few hurdles that the supervised learned classification overcame. The method was also robust to the previously described limitations of using the Kinect v2 sensor. Finally, classifiers were trained in a time-efficient manner (2 minutes maximum) using a commodity graphics processing unit.

VI. Conclusion

In this work, we proposed a method for real-time convolutional neural network classification of static gesture recognition that transcends the virtual and real worlds. Distinctly, the classifier was trained on entirely synthetic samples that underwent a novel data augmentation pipeline for building robust training datasets. Testing network classification performance with separate training dataset variations (kinematic, morphological, rotation, translation) characterized the efficacy of the proposed methods and also provided insights for improving performance of learnt behavior derived from synthetic data. These results have implications with regard to using purely synthetic data to train real-world classifiers in other application domains.

A. Future Work

The results from this support several future research directions. Further exploration and optimization of sample variability can lead to increased recognition robustness for static gestures, while integrating recurrent neural networks (RNN) and virtual environment animations can expand recognition to dynamic gestures as well. Separately, focus can be given to the personalization of gestures and quantifying the process of adding unique gestures to the classification suite.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[3] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," Technical report, Tech. Rep., 2017.

[4] S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data-glove and a speech synthesizer," *IEEE transactions on Neural Networks*, vol. 4, no. 1, pp. 2–8, 1993.

[5] A. Lumini and L. Nanni, "Fair comparison of skin detection approaches on publicly available datasets," *arXiv preprint arXiv:1802.02531*, 2018.

[6] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 423–428.

[7] Y. Iwai, K. Watanabe, Y. Yagi, and M. Yachida, "Gesture recognition by using colored gloves," in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 76–81.

[8] J. Segen and S. Kumar, "Shadow gestures: 3d hand pose estimation using a single camera," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1. IEEE, 1999, pp. 479–485.

[9] L. Bonansea, *Three-dimensional hand gesture recognition using a ZCam and an SVM-SMO classifier*. Iowa State University, 2009.

[10] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 1521–1527.

[11] M. Chen, G. AlRegib, and B.-H. Juang, "Feature processing and modeling for 6d motion gesture recognition," *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 561–571, 2013.

[12] D.-Y. Huang, W.-C. Hu, and S.-H. Chang, "Vision-based hand gesture recognition using pca+ gabor filters and svm," in *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on*. IEEE, 2009, pp. 1–4.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[14] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[15] R. Lionnie, I. K. Timotius, and I. Setyawan, "An analysis of edge detection as a feature extractor in a hand gesture recognition system based on nearest neighbor," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*. IEEE, 2011, pp. 1–4.

[16] S. Marcel and O. Bernier, "Hand posture recognition in a body-face centered space," in *International Gesture Workshop*. Springer, 1999, pp. 97–100.

[17] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.

[18] C.-J. Tsai, Y.-W. Tsai, S.-L. Hsu, and Y.-C. Wu, "Synthetic training of deep cnn for 3d hand gesture identification," in *Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2017 International Conference on*. IEEE, 2017, pp. 165–170.

[19] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*. Springer, 2018, pp. 621–635.

[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 3, no. 4, 2017, p. 6.