Augmented reality data generation for training deep learning neural network

Kevin Payumo[b], Alexander Huyen[a], Landan Seguin[c], Thomas Lu[a], Edward Chow[a], Gil Torres[d]

[a]NASA/Jet Propulsion Lab/California Institute of Technology, Pasadena, CA, USA; [b]Univ. of Calif. Irvine, CA, USA; [c]Georgia Institute of Technology, GA, USA; [d]Naval Air Warfare Center, Point Mugu, CA, USA

ABSTRACT

One of the major challenges in deep learning is retrieving sufficiently large labeled training datasets, which can become expensive and time consuming to collect. A unique approach to training segmentation is to use Deep Neural Network (DNN) models with a minimal amount of initial labeled training samples. The procedure involves creating synthetic data and using image registration to calculate affine transformations to apply to the synthetic data. The method takes a small dataset and generates a high-quality augmented reality synthetic dataset with strong variance while maintaining consistency with real cases. Results illustrate segmentation improvements in various target features and increased average target confidence.

Keywords: Synthetic data, IR image processing, Computer vision, Deep learning, Neural network, Image transformation, Image registration, Augmented reality

1. INTRODUCTION

We explore a method of optimizing data augmentation approaches for deep learning enabled object recognition and segmentation. Deep learning was proven to outperform other algorithms for image segmentation, and a wide variety of deep neural network architectures and datasets were developed over the past several years to utilize its strength [1][2]. However, deep learning approaches require large datasets in order to be generalized and perform well. The quality and size of the dataset are arguably just as important as the optimization of network parameters [3]. One option is to apply data augmentation to significantly enlarge an existing dataset and maximize performance of a deep learning model without spending big effort in collecting many training and validation samples. Data augmentation has been proven to improve image segmentation models and this is especially beneficial when working with small training datasets [4, 5]. On the contrary, increasing the size of a dataset increases the training time. The problem we are interested in is whether the resulting increase in training time is worth the potential improvements. Some efforts have been made to reduce the transformation space in which new data can be generated, such as using expert knowledge to specify worthwhile transformations [6]. Similarly in this study, we use the methods of augmented reality to automatically guide data augmentation in directions that better represent real world dynamics. We expected that this approach would reduce the amount of data required to achieve stronger network performance and reduce the limitations of data collection.

The motivation for this study is due to situations when there is lack of availability of labeled data, which is recommended in large volumes for supervised deep learning. Particularly for smaller players or those new to the field, data might be difficult to collect; hindered by monetary and time expenses. Furthermore, deep learning is susceptible to poor training efficiency, and fully training a deep neural network requires a significant amount of time

especially if computing resources are limited [7]. The rate of progress in this field can be improved by making further progress in data accessibility and training optimization.

1.1 Data Augmentation

Data augmentation is a method for increasing the size of a dataset and the approach has demonstrated performance benefits in deep learning even with synthetic data [8]. A dataset can be augmented using a variety of processing tools, mainly by adding noise to an image, performing geometric transformations, or adjusting the original colors and contrast. As a result, the augmented dataset artificially introduces new information that can improve generalization of the network and prevent overfitting.

We propose a method of using Augmented Reality (AR) for data augmentation. The tracking techniques used in AR allow for guided transformations of our synthetic targets. The superposition of virtual objects onto real backgrounds allow for variation in background information. In the ideal case, we can simulate full videos worth of data with a single synthetic virtual object and then train a deep neural network using the artificial data.

1.2 Augmented Reality

AR virtually integrates computer-generated information onto real-world environments. Much of the emphasis on this research has been placed on developing robust tracking systems and display [9]. State-of-art AR has allowed for a variety of applications in simulation, education, medical, and more [10]. In this paper, the main focus is in using tracking systems to trace the dynamics of an object in one video, and then apply it to the 2D synthetic target frames while superimposing them on real background frames.

For tracking, intensity-based registration methods are used, as they are the most effective for image registration of the low-resolution IR images. We found that feature-based tracking methods are unable to find distinct features in our data and this quickly led to drifting of feature points. Background data only involved images of the sky so this information is less costly to collect. On the other hand, the object information requires the flight of a helicopter and this data is not easily accessible, especially in the required data formats. Previous works on training with synthetic images as a supplement to real images demonstrated promising results in image segmentation models [11]. Thus, we created simple synthetic versions of our target as a more cost and time efficient alternative.

The purpose of the AR research is to provide an efficient tool for generating training samples to the deep learning system for object recognition and segmentation applications.

1.3 Deep Learning for Image Segmentation

Image segmentation requires an understanding of an image at a pixel level. We need to match each pixel to each object class (pixel-wise object recognition problem). Before deep learning dominated the image segmentation problem, pixel-level decision trees [12] and random forests [13] were generally used as classifiers. In 2014, fully convolutional networks were presented and dramatically increased the semantic segmentation accuracy from 75.6% pixel accuracy by conventional support vector machine based method [14] to 85.2% [15] in the SIFT Flow dataset.

Convolutional neural networks use pooling layers to increase the field of view and efficiently take into account the context of the image, while sacrificing the *where* information. However, the *where* information is important in semantic segmentation. To solve this issue, an encoder-decoder architecture with transfer learning from the encoder layers to the corresponding decoder layers (preserving the *where* information) was developed for fast and precise semantic segmentation [16]. Another technique is called the dilated convolution, which allows for aggregating multi-scale contextual information without losing resolution [17].

2. METHODS

2.1 Data Augmentation

Our approach involves using linear transformations to describe how one perspective transforms to another. The strongest type of transformation is the projective transformation, which describes changes in perspective qualities as viewpoints of a scene changes. Projective transformations utilize $\mathbb{R}^3$ to illustrate projective geometries and it can be described by the following relations:

$$\overrightarrow{C_2} = \overrightarrow{C_1} * \boldsymbol{A} \tag{1}$$

$$\boldsymbol{A_{proj}} = \begin{bmatrix} a & b & e \\ c & d & f \\ g & h & i \end{bmatrix} \tag{2}$$

where $\overrightarrow{C_1}$ is a moving target's spatial coordinate position, $\overrightarrow{C_2}$ is the transformed coordinate position, and $\boldsymbol{A}$ is the transformation matrix that brings $\overrightarrow{C_1}$to $\overrightarrow{C_2}$.

Projective transformations were initially of interest because a variety of data has demonstrated significant perspective changes throughout the video. Finding the appropriate parameters for transformation is dependent on obtaining the best correspondence points between two images. This was difficult to achieve with low resolution data and thus for our study we used affine transformations, a subset of projective transformations which are defined as follows:

$$\boldsymbol{A_{aff}} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\boldsymbol{A_{aff}}$ is similar to (2) except g and h, elements of projection, are both equal to 0.

The affine transformation is restricted as a coplanar transformation but is still able to apply a combined effect of translation, rotation, scaling, and shear changes. This combination was sufficient in generating the target data variations we wanted to achieve. Furthermore, it was the most robust approach when used on our low-resolution IR data.

Target data was augmented using image registration from MATLAB's image processing toolbox. A means squares error (MSE) metric is applied to measure to similarity of two frames. MSE quantifies similarity by summing the squared differences of intensity values between two images. An iterative optimization approach called regular step gradient descent optimizer was implemented to minimize the mean squares error. This optimizer acts in accordance with the similarity metric towards the directions of the extrema.

As an example, Figure 1 illustrates the image transformation process. Two different view angles of a helicopter model are shown in Figures (a) and (b). Figure 1 (c) shows that using the Affine transform, the first image in (a) is transformed to the second view angle in (b); Figure 1 (d) shows the pixel difference between the transformed image in Figure 1 (c) and the second view angle in Figure 1 (b). The low intensity area on the helicopter shows the transformation is successful where there is small difference in intensities. The high intensity region in the right side is due to blocked view between the two view angles. We can see that the Affine transform is effective and most of the ship have been matched between the transformed image in Figure 1(c) and the original image in Figure 1 (b).

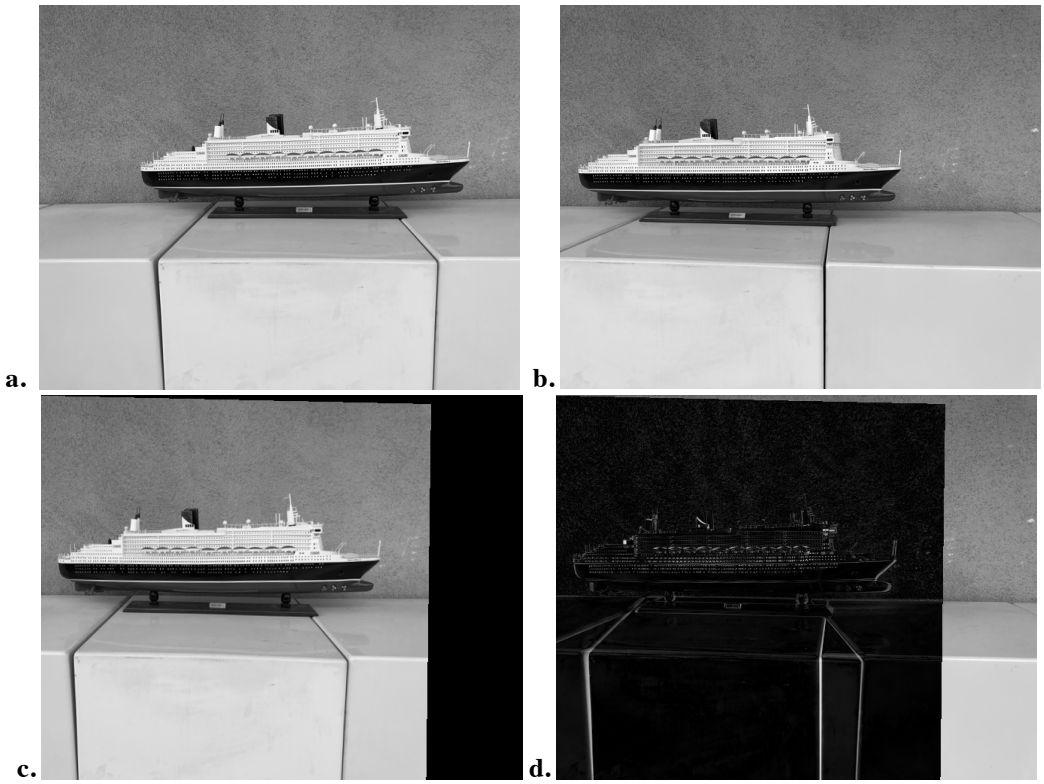
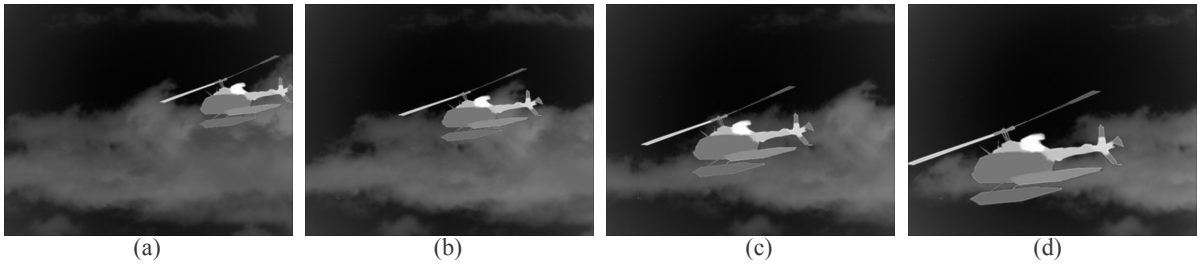


Figure 1: Image transformation: (a) First view angle of a helicopter model; (b) Second view angle of the helicopter model; (c) The first image in (a) is transformed to the second view angle in (b) by using an affine transform; (d) Pixel difference between the image in (b) and the transformed image in (c). The two views are matched using Affine transformation.

External videos that contain an object demonstrating a desired motion were used as reference to retrieve the transformation matrices to apply to the synthetic data. These videos do not necessarily have to be of the same style representing the model; only the dynamic properties of the object need to be representative. Once the synthetic data has been transformed, the sequence of frames can be superimposed onto a sequence of background images as shown in Figure 2.



(a) (b) (c) (d)

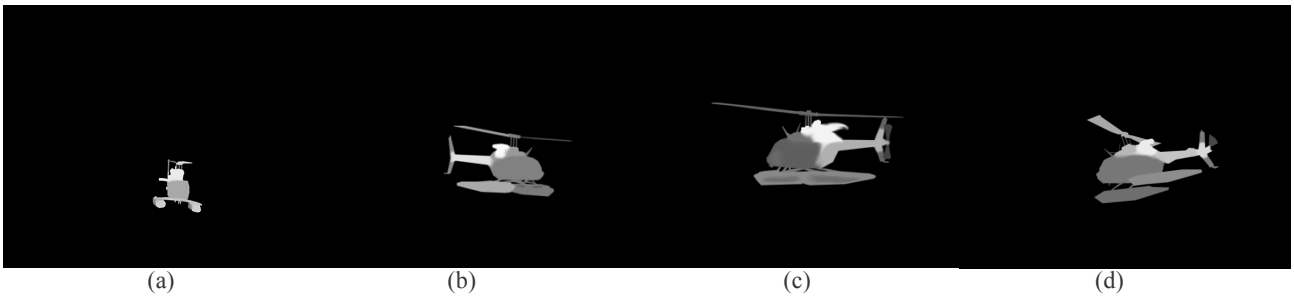
Figure 2. Example of an augmented reality synthetic video sequence created through successive transformations and superimposing the resulting images onto a sequence of background frames.

2.2 Synthetic Data/Augmented Reality

As a baseline for applying augmented reality, initial target data is generated through two different sources:

1. 2D computer-generated (synthetic) target images created in GNU Image Manipulation Program (GIMP), see Figure 3.
2. Extracted targets from real videos through various segmentation methods such as manual segmentation, grabcut (or GraphCut), thresholding, and roughly trained network models.

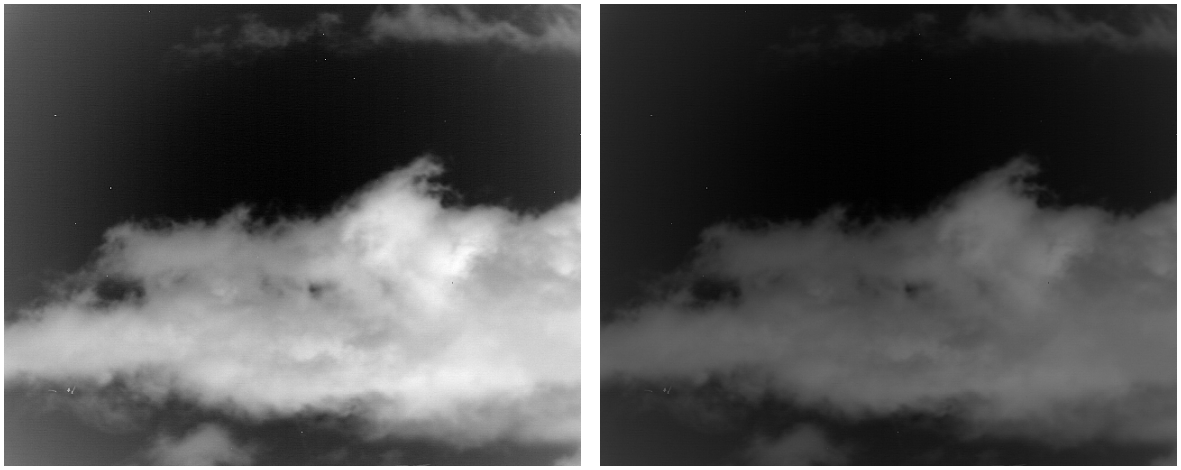
These different target data sources are designed to demonstrate training dependence on object texture. Augmented reality deals with computer-generated objects and in many cases, the textures in these graphics can easily help us identify an object as artificial. Our synthetic images have minimal textures across the targets, and are rather uniform intensities for a majority of the target regions. Our results aim to illustrate a lower boundary for network performance when training on augmented reality data.



(a) (b) (c) (d)

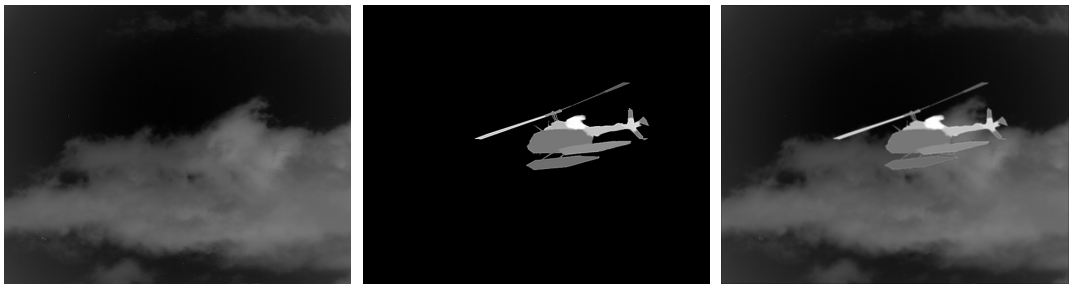
Figure 3. Synthetic images (a) – (d) generated in GIMP with limited intensity variation and modeled after the real objects.

Various background videos are collected related to the object model. Background frames are preprocessed using the *imadjust* function in MATLAB's image processing toolbox, as shown in Figure 4. The augmented objects are superimposed onto the real background frames, as illustrated in Figure 5.



(a) (b)

Figure 4. Preprocessing of the background frame to be qualitatively more representative of the real videos (a) Original background image; (b) The preprocessed background frame.



(a) (b) (c)

Figure 5. Augmentation process: (a) Preprocessed background frame, (b) Synthetic target frame, (c) Superposition of Target and background frames.

2.3 Pix2pix Neural Network Model

Pix2pix [18] is a deep neural network (DNN) that features conditional generative adversarial networks to perform image-to-image translation tasks. This deep neural network model learns to map an input image to an output image, and to minimize a loss function that adapts to the training data to generate output that best represents the desired outcome. Traditionally, loss functions must be selected using specialized expertise about the data to produce acceptable results. Different tasks require significantly different loss functions, which affect the accuracy of the training and quality of the neural network's output. Conditional generative adversarial networks (cGAN) learn a structured loss function that prioritizes minimizing the difference between the neural network output and objective.

This deep neural network model is trained on a variety of synthetic and real data. Augmented synthetic data is used to investigate whether real image features could be abstracted by cGANs by training on this synthetic data. New synthetic training data is generated quickly and efficiently with minimal manual work to represent the existing set of real image data. Different combinations of synthetic and real images have been used to investigate the effects from augmented reality data on the accuracy of segmentation tasks.

2.4 Training Deep Neural Networks

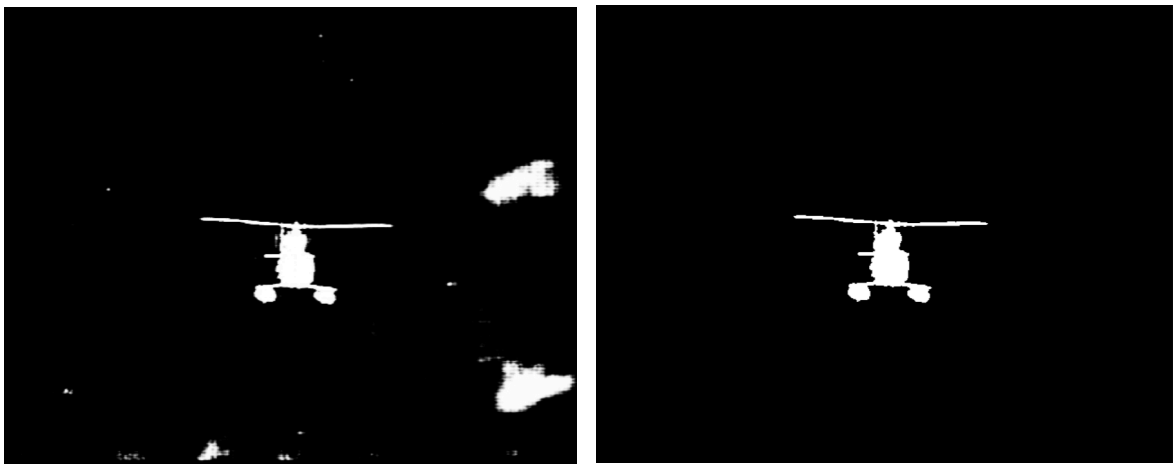
A variety of training datasets are generated to compare performance. These datasets are prepared as follows:

1. Complete Real Dataset: All of existing manually drawn ground truths and training data to compile a set of 154 training samples.
2. Initial Real Dataset: Object data was extracted from the real dataset and then superimposed onto real background frames to create an initial set of 10 training samples.
3. Initial Synthetic Dataset: Synthetic object data are modeled after the initial real dataset and superimposed onto real background frames to create a initial synthetic set of 10 training samples.
4. Multiple Augmented Real Datasets: Initial real data was augmented and superimposed onto real background images. 20 images were added every training run.
5. Multiple Augmented Synthetic Dataset: Initial synthetic data was augmented and superimposed onto real background images. 20 images were added every training run.

Training sets involved with 4 and 5 are specifically augmented based on the performance of the previous trainings. We performed a series of trainings, increasing the training sample size by 20 images at a time. The 20 images included in each set are primarily chosen and created based off segmentations that were unsatisfactory in the previous test runs. The original image resolution is 512 x 640 pixels however they are resized for training with input and output resolutions of 512 x 512 pixels. All training is done using a NVIDIA GTX1080 GPU.

2.5 Post-Processing

The raw segmentation output from the DNN usually contains some noise so a computer vision library, *openCV,* is used to perform post-processing on the DNN output. Since the segmentation was done on videos, we took advantage of assumptions about neighboring frames. The key assumption is that the background is more dynamic than the target with respect to the camera's view. A subset of video frames (every 25 frames) were taken and created bounding boxes around significant contours. The boxes were used to count the number of detections that occurred in each box throughout the whole video. Since the object is usually within the same region in most of video (the camera is tracking the object), the highest scoring bounding box can be assumed as the one that contains the true object segmentation. Once the object bounding box is detected, we return to the frame that the bounding box was taken from, and use the box as a reference to process unwanted noise in the neighboring frames. The bounding box is then resized and repositioned to the detected target and the sequence continues to the first and last frames in the video. Figure 6 shows the effect of the post-processing step that clouds in the background are removed.



(a) (b)

Figure 6. Example of post-processing of a segmentation result. (a) DNN segmentation output. (b) Post-processed output. Unwanted cloud segmentations were removed automatically with the post-processing algorithm.

3. RESULTS AND ANALYSIS

3.1 Measuring Performance

To measure the network performance we focus on the XOR accuracy, which is defined as follows:

$$A_{XOR} = \left(1 - \frac{\sum_{i=1}^{n} p_{GT}(x_i, y_i) \oplus p_{DNN}}{\sum_{i=1}^{n} p_{GT}(x_i, y_i)}\right) * \ 100 \tag{4}$$

where, A_{XOR} is the percent accuracy of the segmentation, n is the number of coordinate pairs in the image, $p_{GT}(x_i, y_i)$ is the pixel value at position (x_i, y_i) of the binary ground truth (GT), $p_{DNN}(x_i, y_i)$ is the pixel value at position (x_i, y_i) of the binarized DNN segmentation output after post-processing. It is important to note that edges are ambiguously defined due to the intensity gradient between the target and background. This gradient makes it difficult to consistently define the edge and a difference in 1-pixel width can contribute to a notable portion of error.

Two different testing sets are prepared. One set contains 154 images and includes test samples that are difficult to simulate with our proposed method of generating AR data. Consequently, the training data set will have limited representation of several images in this testing set. The 116-image training set is designed to measure how the network performs on samples that are more closely representative of the training sets and is expected to have slightly higher accuracies. The average accuracy was calculated across all images in each set.

3.2 Augmented Reality/Synthetic Data Training

The training set includes an initial set of synthetic target frames, which are created using GIMP and binarized to create their GTs. This set of data is transformed using the proposed method to augment the training dataset. Table 1 shows the results of test accuracy vs number of synthetic images added to the training set. The DNN performs better when more augmented synthetic images are used to train the DNN. Figure 7 shows the graph of the testing accuracy vs number of training samples using the 116-image set and 154-image set.

Table 1: Synthetic data training average test accuracy results on the 116-image test set and the 154-image test set.

# of Images in Training Set	10	30	50	70	90	110
116-Image Set Test Accuracy	89.30%	90.86%	90.84%	91.17%	91.99%	92.21%
154-Image Set Test Accuracy	86.16%	89.12%	89.75%	89.82%	90.90%	91.30%

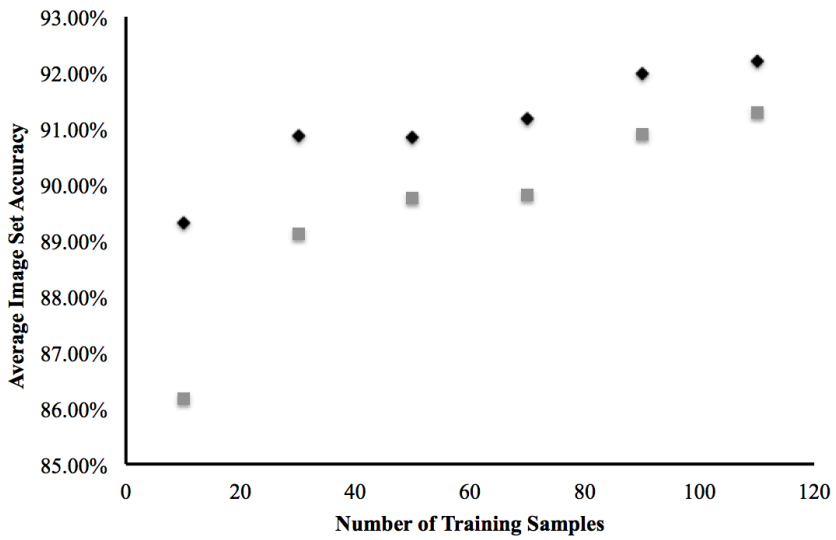


Figure 7. Average image set accuracy is plotted against the number of training samples for synthetic-target augmented images. The black diamonds represent the accuracy on the 116-Image set and the grey blocks represent the accuracy on the 154-Image set.

The series of trainings begins with fairly low with an accuracy of 86.16% on the 154-Image Set and 89.30% on the 116-Image set. As expected, the segmentation accuracy gradually increases as more training images are continuously added to the training set. The 154-Image set has lower average accuracies as is expected but both the 154-Image set and 116-Image set followed similar trends as new training samples were included. The gaps in performance between the two sets may be due to the different texture in distinguishing foreground objects to more complex backgrounds.

3.3 Real Augmented Data Training

This training set includes an initial set of real object frames, which are extracted from the real images by using their respective ground truths. This set of data is transformed using the proposed method to augment the training dataset. Table 2 shows the results of test accuracy vs number of real augmented images added to the training set. The DNN performs better when more augmented real images are used to train the DNN. Figure 8 shows the graph of the testing accuracy vs number of training samples using the 116-image set and 154-image set.

Table 2: Transformed real data training average accuracy results on the 116-image test set and the 154-image test set.

# of Images in Training Set	10	30	50	70	90	110
116-Image Set Accuracy	88.34%	91.05%	89.88%	91.50%	91.79%	92.78%
154-Image Set Accuracy	88.22%	89.76%	89.39%	91.29%	91.35%	91.85%

Again, the network trained on 10-images starts off with relatively low accuracy in the segmentations. As images are added, the segmentation results tend to gradually improve. However, the 50-image training set produces unexpected results and performs less accurately than the previous 30-image training set. Still, the accuracy tends to increase with the greatest improvement in the training is an XOR accuracy of 92.78% (+4.44% from initial results) on the 116-image training set. In contrast with the gaps observed in synthetic training set, the real augmented data results exhibit several training runs in which the 154-Image set accuracy is more comparable to the 116-Image set accuracy. On the other hand, the results still demonstrate that the 116-Image set had higher accuracy as expected.

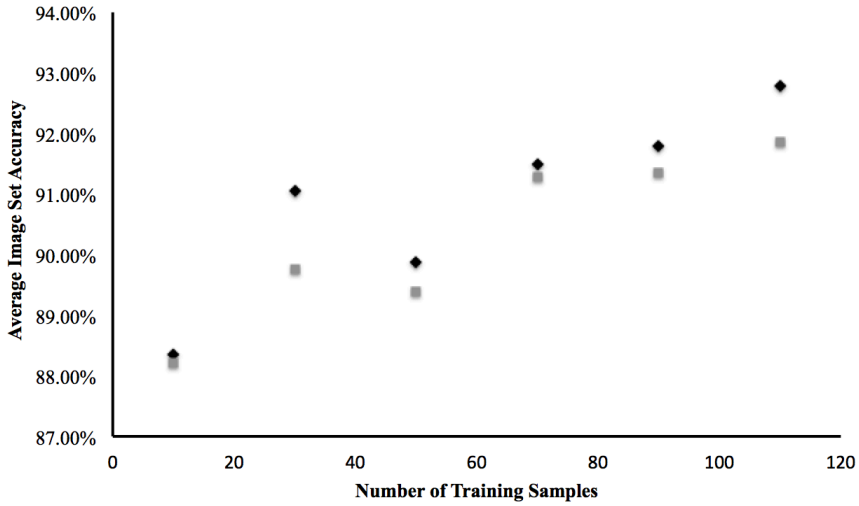
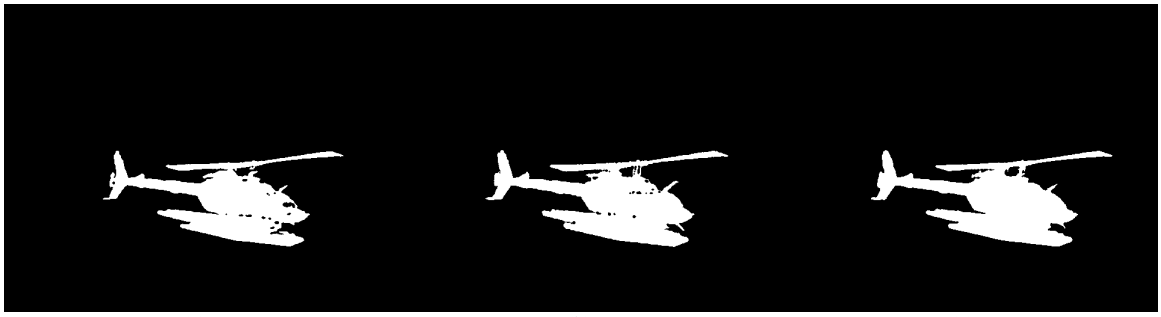


Figure 8. Average image set accuracy is plotted against the number of training samples for real-target augmented images. The black diamonds represent the accuracy on the 116-Image set and the grey blocks represent the accuracy on the 154-Image set.

Figure 9 shows the progression of improved accuracy as the training samples are added to the training set. Consistent improvements are expected as images are added. However, in the 50-image training set, the accuracy decreases. The cause for this may be due to set of images added to the training sets during these runs. More training images are added to the training data set based on the previous training's performance, but it is difficult to guarantee that the new images will be beneficial to the network training.

When comparing these trainings to the synthetic data, we see there is less than a 1% difference between the two types. We expect the difference between them to be larger since the synthetic targets lacked detailed texture. This similarity in the performance could be accounted for by the fact that the background data came from real videos. Additionally, when looking through this set's segmentation results, we found that one image scored a XOR Error of 61.07%. However, the testing set sizes are large enough such that outlier segmentation results like this one will not change the mean XOR error significantly.



(a) (b) (c)

Figure 9. Example of progression of segmentations as more images are included: (a) Result from initial training set (10 image training set), (b) Result from second retraining iteration (50 image training set), (c) Result from fourth retraining iteration (90 image training set).

4. DISCUSSION

Applying AR techniques to generate datasets to train deep neural networks is a promising approach for automated image segmentation. Our results demonstrated that increases in dataset size led to improvements in segmentation accuracy. More optimization needs to be done so that the accuracy converges to a maximum range using this method. The ground truths were generated manually and edges are inconsistently defined frame-to-frame. They may

be ill defined within 1-5 pixels due to the intensity gradient observed as target edges transition into background regions. As reference, we are able to achieve a 95.73% average accuracy (3.88% higher than our real augmented dataset) using our complete ground truth set of 154 images. Note that these results were achieved using training data that is also in the validation set. Nonetheless, we aim to achieve more similar results using our proposed method.

There are a variety of ways to improve the proposed approach. Currently, there is still more than desired manual work required when preparing our datasets. In our study, we added training samples based on the performance of the previous network but in the ideal case, we should not have to check so often to generate a satisfactory training set. Furthermore, our current method of transformation is sensitive to large differences between frames and needs to be optimized for each video. Multiple successive transformations on a single target leads to blurring of edges and textures which diminishes the quality of our data. Furthermore, depending on the pair of target and background, there may be unnatural looking gradients along the target edges. These two effects can place limitations on the precision of the segmentations and requires us to review training data constantly. Currently, the synthetic AR data achieves similar accuracies as the real augmented data. In the final training runs with 110 images, the synthetic AR trainings performed within 1% that of the real augmented data, further supporting the idea of using synthetic data to train deep learning models for image segmentation.

Our study explored the capabilities of DNNs for image segmentations when trained on augmented reality data and should be treated as a lower boundary limit for segmentation of single class low-resolution IR images. Future works include creating 3D models of our targets and implementing state-of-art augmented reality display and tracking techniques. Photogrammetry is a promising approach for 3D reconstruction of objects but it will require higher resolution data. The use of 3D modeling should better the process of automatic synthetic data generation. Additionally, the availability of higher resolution data will allow us to take advantage of better methods of transformations, such as projective transformations, for our task.

To further optimize segmentation results we are interested in applying texture synthesis to maximize the quality and representation of our training data. Texture synthesis will enable us to use a wider variety of databases by converting the source data into the appropriate style for our model. Our initial test runs of texture synthesis using a similar network architecture of our segmentation model has shown some promising implications; at least qualitatively. We aim to apply texture synthesis to binary data and then train a segmentation based deep network with the processed training data.

5. CONCLUSIONS

AR is a useful method for creating data for DNN training as it can automatically guide data augmentation by tracing the real-world dynamics of an object in a reference video. The results from training a DNN using AR for dataset augmentation has demonstrated effectiveness in precision segmentation of IR images. As more images were added to the datasets, network performance increased and further supports the concept of using data augmentation to improve the quality of training datasets. For single class, low-resolution IR images, our AR approach score achieved an accuracy of 92.21% when using synthetic targets and 92.78% when using real targets. The 110 image synthetic training dataset performed within 1% similar accuracy as the 110 image real dataset, which also validates the effectiveness in training a DNN with synthetic data. The results of this study suggest that AR can be applied as a convenient tool for training a DNN for precision IR image segmentation.

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology and was supported in part by the U.S. Department of Defense, Test Resource Management Center, Test & Evaluation/Science and Technology (T&E/S&T) Program under NASA prime contract NAS7-03001, Task Plan Number 81-12346.

REFERENCES

1. Garcia-Garcia, A., S. Orts-Escolano, S.O. Oprea, V. Villena-Martinez, and J. G.-R., "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv:1704.06857,* (2017).
2. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L., "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *arXiv:1606.00915*. (2016).
3. Sun, C., Shrivastava, A., Singh, S., & Gupta, A., "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," (2011).
4. Taylor, L., & Nitschke, G., "Improving deep learning using generic data augmentation," *arXiv Preprint arXiv:1708.06020*, (2017).
5. Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., & Jin, Z., "Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation," *arXiv:1601.03651*, (2016).
6. Vasconcelos, C. N., & Vasconcelos, B. N.,"Convolutional Neural Network Committees for Melanoma Classification with Classical And Expert Knowledge Based Image Transforms Data Augmentation", *arXiv preprint arXiv:1702.07025*, (2017).
7. Wu, R., Yan, S., Shan, Y., Dang, Q., & Sun, G., "Deep Image: Scaling up Image Recognition," *arXiv:1501.028768*, (2015).
8. Rajpura, P., Goyal, M., Bojinov, H., & Hegde, R., "Dataset Augmentation with Synthetic Images Improves Semantic Segmentation.," *arXiv:1709.00849*, (2017).
9. Zhou, F., Been-Lirn Duh, H., & Billinghurst, M. (n.d.). AP 9. "Trends in AR Tracking, Interaction and Display: A Review of Ten Years of ISMAR", *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality,* (2008).
10. Noh, Z., Sunar, M. S., & Pan, Z., "A review on augmented reality for virtual heritage system," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, (2009).
11. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M., "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016).
12. Shotton, J., M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, (2008).
13. Montillo, A., J. Shotton, J. Winn, J. E. Iglesias, D. Metaxas, and A. Criminisi, "Entangled decision forests and their application for semantic segmentation of CT images," *Inf. Process. Med. Imaging*, vol. 22, pp. 184–196, (2011).
14. Tighe, J., and S. Lazebnik, "Finding Things: Image Parsing with Regions and Per-Exemplar Detectors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, (2013).
15. Shelhamer, E., J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, (2017).
16. Ronneberger, O., P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Lecture Notes in Computer Science*, pp. 234–241, (2015).
17. Yu, F., and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122,* (2015).
18. Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv:1611.07004*, (2017).