HESCNET: A SYNTHETICALLY PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK FOR HUMAN EMBRYONIC STEM CELL COLONY CLASSIFICATION

Adam Witmer[1,⋆] *and Bir Bhanu*[1,2,†]

[1]Dept. of Bioengineering, University of California, Riverside, CA, 92521
[2]Dept. of Electrical and Computer Engineering, University of California, Riverside, CA, 92521

Email: ⋆awitm001@ucr.edu, †bhanu@cris.ucr.edu

ABSTRACT

This paper proposes a method for improving the results of deep convolutional neural network classification using synthetic image samples. Generative adversarial networks are used to generate synthetic images from a dataset of phase-contrast, human embryonic stem cell (hESC) microscopy images. hESCnet, a deep convolutional neural network is trained, and the results are shown on various combinations of synthetic and real images in order to improve the classification results with minimal data.

***Index Terms*—** Video Bioinformatics, Computer Vision, Image Processing, Deep Learning, Generative Adversarial Networks

1. INTRODUCTION

Video bioinformatics is crucial for the analysis of temporal stem cell behavior in time-lapse microscopy images [1]. The behavioral and morphological changes observed during live stem cell culture are indicative of growth and health status and can be used as features for the classification of cell colonies. Normally, videos taken over hours or days must be manually annotated and quantified by the experimenter. By-hand analysis takes many months, and is unreliable because of experimenter bias. Standardization of these processes is necessary for the reliability and repeatability of experimental analysis. Computer vision and image processing algorithms have been applied to this effect [2, 3, 6].

More recently, deep learning algorithms have been applied to biological datasets to perform a variety of tasks including image classification [4, 5, 7, 8]. Biological experimentation does not always produce the large amount of data required to train these networks effectively, and overfitting can prohibit the use of these algorithms for the intended task. Data augmentation techniques such as image cropping, flipping, rotating, and scaling provide some relief from overfitting, but, in situations with limited data, can be ineffective even for smaller deep-networks. When performing expensive biological experiments (in terms of time, money, and resources), it is impractical to reserve any portion of the dataset for network training, as every data point is necessary for accurate calculation of experimental outcomes.

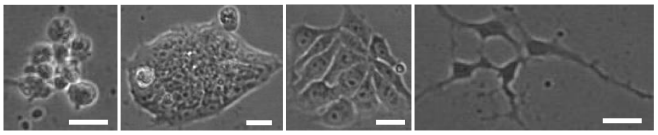

Fig. 1. Examples of cell colony images with scale bars for proportion. From left to right, Debris, Dense, Spread, Differentiated.

Therefore, this work proposes the use of synthetic data for the expansion of an experimental dataset for training deep convolutional neural networks. A large scale dataset is established using Generative Adversarial Networks (GAN) [12] to train a convolutional neural network (CNN) on various combinations of synthetic and real data including mixing datasets, and pre-training as a weight intialization strategy before fine tuning the network with real data.

2. RELATED WORK AND CONTRIBUTIONS

2.1. Related Work

Automated image processing and computer vision methods have been employed extensively in microscopy image analysis. Algorithms exist for the quantification and classification of cells and cell colony images using various heuristic and machine/deep learning features. Zahedi, et al. [3] analyze the health of human embryonic stem cell colonies under toxic

This work is made possible by the NSF IGERT program in Video Bioinformatics (DGE 0903667) and TRDRP (Award ID: 27DT-0007). The contents of the information do not reflect the position or policy of US Government.

The authors would like to thank Dr. Prue Talbot and Barbara Davis of the Talbot laboratory (http://talbotlab.ucr.edu) for providing the experimental data that were used in the project.

conditions using a combination of heuristic-morphological (area, aspect-ratio, protrusions, solidity) and dynamic (motility, growth rate) features. They track colonies in time-lapse, phase-contrast microscopy images over a 48-hour period and use temporal behavior analysis to classify colonies as healthy, unhealthy, or dying with an accuracy of 96%. This method employs an SVM classifier to categorize image samples, but requires the design of hand-crafted features that do not generalize to multiple cell types. In contrast, deep learning features are a powerful method of incorporating multiple data classes into a single model, but require a large data set for iterative training to be effective. Traditional data augmentation techniques are effective for increasing variation during training, but are still subject to limitations in available data. More recently, the supplementation of small data sets with representative synthetic samples has shown promise as method of improving classification results.

Xie, et al. [8] perform cell counting in fluorescent images using a convolutional regression network trained only on synthetic data. They circumvent manual annotation of cell colonies by training a network to localize fluorescently labeled cell nuclei via down-convolutional feature extraction and symmetrically up-convolutional pixel-wise classification. They apply their network to a variety of data-sets including synthetically validated fluorescent images, and manually annotated gray-scale histology sections with an average error of 2.9% for their cell counting task. While their method is a successful implementation of synthetic data for training a neural network feature classifier, their fluorescent dataset is relatively easy to replicate in a representative manner. More complex datasets, such as data with low contrast and high texture cannot be easily simulated, requiring a more sophisticated method than manual cloning. One potential method for replicating complex datasets is with Generative Adversarial Networks. These networks use unsupervised neural network training data to generate synthetic data that is representative of the experimental dataset distribution. Therefore, the following contributions are proposed for this work.

2.2. Contributions

- Generated a library of synthetic biological image data using Generative Adversarial Networks.
- Trained a convolutional neural network using generated data to supplement minimal dataset. The results are shown for various combinations of synthetic and real data.

3. TECHNICAL APPROACH

3.1. Data, Classes & Ground-Truth

Data for this project comes from the developmental toxicology laboratory of Dr. Prue Talbot, in the Departement of Cellular, Molecular and Developmental Biology at the University of California, Riverside. Induced pluripotent stem cells (iPSC) expressing the Huntington's disease phenotype are cultured under exposure to nicotine solution. Nicotine has been shown to have a neuro-protective effect on patient's with neurodegenerative diseases in a clinical setting [9–11]. Stem cell models are effective in determining the early developmental effects of chemical exposure because they accurately replicate the embryonic growth cycle in vitro.

Cells are cultured in the Nikon Biostation CT (Nikon Instruments) and imaged using phase-contrast optical microscopy once an hour for 48-hours in order to observe temporal colony behavior. The resulting dataset is comprised of 15 videos of 48 frames each (720 images), 2098 x 2098 pixel resolution. Visual observation of these time-lapse videos reveals colony morphology changes on the cellular level that translate to variations in colony texture. Within each frame, multiple colonies exist at various stages of growth and differentiation, and can be compartmentalized into four morphological classes: *Dense*, having low cell cytoplasm to nuclear area, tightly packed with no clear cell boundaries, indicative of undifferentiated, pluripotent stem cell colonies; *Spread*, having high cytoplasm to nuclear area, larger individual cells within the colony with clear, high intensity cell boundaries, indicative of a downstream progenitors; *Debris*, individual, or small aggregates of rounded, high intensity cells having a bubble like appearance, indicative of unhealthy or dying cells; and *Differentiated*, having a more clear neural morphology, with dark cell body and long, thin protrusions indicative of axon development. Figure 1 contains representative examples of these colony classes.

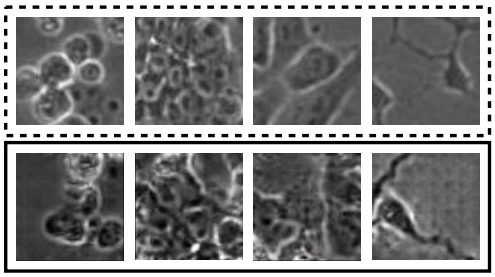

Fig. 2. Real (top) and synthetic (bottom) image patch examples. From left to right, 64 x 64 Debris, Dense, Spread, Differentiated samples.

Table 1. Data Breakdown for Four Classes and Number of Images Associated with Various Training Schemes

Class	# Samples	10%(total)	80%(total)	50%(total)
Debris	3587	347	2884	1813
Dense	3934	372	3137	1980
Diff	656	62	509	342
Spread	10506	1087	8413	5204

Table 2. hESCnet Network Architecture

Module	Size Dim (In/Out)	Feat. Maps (In/Out)
Conv2d	64/64	3/64
Conv2d	64/64	64/64
MaxPool2d	64/32	64/128
Conv2d	32/32	128/128
Conv2d	32/32	128/128
MaxPool2d	32/16	128/128
Conv2d	16/16	128/256
Conv2d	16/16	256/256
Conv2d	16/16	256/256
Conv2d	16/16	256/256
MaxPool2d	16/8	256/256
Conv2d	8/8	256/512
Conv2d	8/8	512/512
Conv2d	8/8	512/512
Conv2d	8/8	512/512
MaxPool2d	8/4	512/512
Linear	4/1*4096	512*4*4/4096
Linear	1*4096/1*4096	4096/4096
Linear	1*4096/1*4	4096/4

In order to effectively exploit these textural differences, cell colonies are first segmented from the flat background of the culture medium using a standard morphological segmentation algorithm (entropy filtering, image binarization via Otsu thresholding, ROI opening, hole filling, small object removal). This results in a binary map of cell colony location, from which colony bounding boxes are cropped and individually labeled by-hand into the four classes. The image breakdown of the dataset resulting from ground-truth labeling, along with the number of images used in the various experiments by percentage split are detailed in Table 1. Accurate analysis of the biological experiments necessitates that all of the data be used for testing the network. Unfortunately, a large portion of the dataset must be used to train the neural network effectively. Generative adversarial networks are used to supplement real data with synthetic data samples for network training, so that the majority of the experimental data samples can be used during the testing phase.

3.2. Generative Adversarial Network

A recent method of creating synthetic data is the Generative Adversarial Network (GAN) [12]. A convolutional implementation of this deep learning based algorithm, deep convolutional generative adversarial network (dcGAN), utilizes two convolutional neural networks to generate synthetic image data from an unlabeled dataset [13]. One network, the *Generator*, takes a noise vector of length 100 as input, and produces a 2D tensor of size 64 x 64 via convolutional transpose upsampling. The second network, the *discriminator*, takes the generated image as input, and determines the probability that the input came from the real data, versus the probability distribution of the generator. The algorithm then updates the weights of both networks with respect to the output of the discriminator via back propagation.

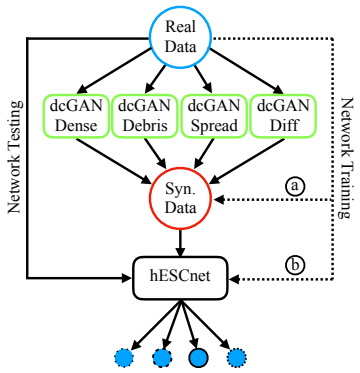

Fig. 3. Network Training/Testing Diagram. Subclasses of the real dataset are used to train separate dcGAN's, from which synthetic images are compiled to create a synthetic dataset. Various combinations of synthetic and real data are used to train the network (a), as well as comparing to the network trained only on real images (b).

During training, the objective of the generator is to minimize the probability that the generated sample, z, is drawn from the generated distribution, p_z. In contrast, the discriminator attempts to maximize the probability that a given sample, x, comes from the real data distribution, $p_{data(x)}$. The network trains using this minimax decision rule (Equation 1), where $\mathbb{E}$ is the expected value formula, until the training losses of the networks equilibriate

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log\ D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

The result of this training is a generator network that is able to produce synthetic images that lie within the distribution of the real dataset, using only a random Gaussian noise vector as input. Four dcGAN's are trained on separate datasets containing only one data type. Each network is trained on random 64 x 64 image patches extracted from the real colony crops. Image patches are generated for each individual class and used to supplement the real dataset for CNN training. Examples of real and synthetic image patches for each class can be seen in Figure 2. The general overview of this approach is shown in Figure 3.

3.3. Deep Convolutional Neural Network

hESCnet, a deep CNN, is trained on multiple combinations of real and synthetic data, including being pre-trained on syn-

Table 3. hESCnet Parameters

hESCnet Configuration	Train %	Test %	Validation %	Synthetic Images/Class	Pretrained Synthetic
a	80	10	10	0	No
b	10	45	45	0	No
c	10	45	45	0	Yes
d	10	45	45	500	No

Table 4. hESCnet Classification Results

Network Config.	Accuracy $\pm$ Std	True Positive Rate $\pm$ Std	ROC (AUC) $\pm$ Std	Train/Time (min) $\pm$ Std	Min. Train Loss
a	$\mathbf{0.857 \pm 0.007}$	$\mathbf{0.897 \pm 0.008}$	$\mathbf{84.600 \pm 0.012}$	58.621 ± 0.769	0.167 ± 0.010
b	0.832 ± 0.011	0.871 ± 0.005	81.740 ± 0.007	29.453 ± 0.099	0.193 ± 0.014
c	0.854 ± 0.004	0.864 ± 0.007	81.330 ± 0.116	$\mathbf{29.376 \pm 0.1145}$	0.185 ± 0.022
d	0.827 ± 0.009	0.875 ± 0.004	83.480 ± 0.007	43.517 ± 9.734	$\mathbf{0.027 \pm 0.005}$

thetic data before being refined using a small portion of the real dataset. The overall architecture of the CNN is described in Table 2.

The network is based on the VGG network [14], but takes an image of size 64 x 64 as input. All convolutional kernels are of size 3 x 3, with stride and padding parameters of (1,1). Rectified linear unit activation, and batch normalization follow every convolutional module and dropout layers are used after each fully connected, linear module. A learning rate of 0.005 and weight decay of 0.0001 were experimentally determined for the stochastic gradient descent training algorithm. The cross entropy loss criterion is employed and each network variation is trained for 200 total epochs, decreasing the learning rate by a factor of 10 after 100 epochs. All networks are trained in parallel, across 2-NVIDIA GTX 1080Ti GPU's.

Four variations of the dataset, detailed in Table 3, are used to train hESCnet. During training, random 64 x 64 crop patches are taken from the real input images and randomly flipped horizontally and vertically during each iteration. The synthetically generated patches require no pre-processing before training. All networks are tested only on real, experimental image patch samples. The average results of network testing using 10-fold cross-validation are detailed in the following section.

4. RESULTS & DISCUSSION

hESCnet testing results are outlined in Table 4. Network results are quantified using the overall accuracy of positively classified samples, true positive rate (TPR), and area under the receiver operating characteristic curve (ROC-AUC), as a measure of classifier robustness. While hESCnet(a) produces the highest accuracy, TPR, and ROC-AUC, it represents the least optimal method of training, using 80% of the valuable experimental data for training.

Conversely, hESCnet(b-d) use only 10% of experimental data for training, and perform slightly worse in the accuracy metrics, but take less time to train and produce similar minimum training loss statistics as hESCnet(a). While training the neural network solely on large sets of synthetic data results in network overfitting and does not generalize well to the real dataset, the addition of 500 synthetic images per class to the configuration of hESC(b) during the training (hESCnet(d)) increases the true-positive rate of classification as well as the robustness of the associated feature classifier. These findings indicate that the addition of synthetic data to a small training-dataset increases the classification accuracy of the convolutional neural network, and that synthetic image features generalize to real data in small proportions.

5. CONCLUSIONS

This work addresses the need to supplement small biological image dataset during neural network training. While deep learning methods present the best opportunity for increasing classification accuracy of the associated task, these data expensive networks require large datasets for sufficient feature extraction. Many times, the monetary, time, and logistical expenses of biological experimentation prevent the collection of large scale image datasets. In addition, the training data initially applied to network training cannot be re-used for testing of the same network, thus removing a large portion of valuable data from experimental quantification.

Here, a neural network trained on a combination of synthetic and real image data, hESCnet, is proposed. Generative Adversarial Networks are employed for the generation of synthetic data within the real data distribution. Classification results and the ROC-AUC classifier metric reveal that the supplementation of a small experimental dataset with synthetic images improves network performance. This work represents a promising application of neural network training for biological tasks with limited datasets.

6. REFERENCES

[1] P. Talbot, N. zur Nieden, S. Lin, I. Martinez, B. Guan, and B. Bhanu, "Use of video bioinformatics tools in stem cell toxicology," *Handbook of Nanotoxicology, Nanomedicine and Stem Cell Use in Toxicology*. pp. 379-402, 2014.

[2] B.Guan, B. Bhanu, P. Talbot, and S. Lin. "Bio-driven cell region detection in human embryonic stem cell assay." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11.3, pp. 604-611, 2014.

[3] A. Zahedi, V. On, S. Lin, B. Bays, E. Omaiye, B. Bhanu, and P. Talbot. "Evaluating cell processes, quality, and biomarkers in pluripotent stem cells using video bioinformatics." *PLOS one*, vol. 11.2, 2016.

[4] F. Buggenthin, F. Buettner, P. Hoppe, M. Endele, M. Kroiss, M. Strasser, M. Schwarzfischer, D. Loeffler, K. Kokkaliaris, O. Hilsenbeck, T. Schroeder, F. Theis, and C. Marr. "Prospective identification of hematopoietic lineage choice by deep learning." *Nature Methods*, vol. 14, pp. 403-406, 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: convolutional networks for biomedical image segmentation." *MICCAI*, pp. 234-241, 2015.

[6] T. Perestrelo, W. Chen, M. Correia, C. Le, S. Pereira, A. Rodrigues, M. Sousa, J. Ramalho-Santos, and D. Wirtz. "Pluri-IQ: quantification of embryonic stem cell pluripotency through an image based analysis software." *Stem Cell Reports*, vol. 9, pp.1-13, 2017.

[7] D. Van Valen, T. Kudo, K. Lane, D. Macklin, N. Quach, M. DeFelice, I. Maayan, Y. Tanouchi, E. Ashley, and M. Covert. "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments." *PLOS Computational Biology*, vol. 12. 11, 2016.

[8] W. Xie, A. Nobel, and A. Zisserman. "Microscopy cell counting and detection with fully convolutional regression networks." *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2016.

[9] M. Quik. "Smoking, nicotine and Parkinson's disease". *TRENDS in Neurosciences*, 2004.

[10] A. McGregor, J. Dysart, M. Tingle, B. Russell, R. Kydd, and G.Finucane. "Varenicline improves motor and cognitive symptoms in early Huntington's disease." *Neuropsychiatric Disease and Treatment*, 2016.

[11] M. Tariq, H. Khan, I. Elfaki, S. Al Deeb, and H. Al Moutaery. "Neuroprotective effect of nicotine against 3-nitroproionic acid(3-NP)-induced experimental Huntington's disease in rats." *Brain Research Bulletin*, 2005.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets." *Advances in Neural Information Processing Systems*, vol 27. pp 2672-2680, 2014.

[13] A. Radford, L. Metz, and S. Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *ICLR*, 2016.

[14] K. Simonyan, and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." *ICLR*, 2015.