UNIVERSITY OF CALIFORNIA
RIVERSIDE

Classification of Stem Cell Microscopy Images Using Deep Learning

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Bioengineering

by

Adam John Witmer

June 2024

Dissertation Committee:

Dr. Bir Bhanu, Chairperson
Dr. Prue Talbot
Dr. Boris Hyle Park

The Dissertation of Adam John Witmer is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am very grateful for the guidance, support and patience of my advisor, Dr. Bir Bhanu, who encouraged me to persevere; to Dr. Prue Talbot, who was instrumental in the genesis of my thesis and generous with her time and energy in our collaborative research projects; and to Dr. Barbara Davis, who conceived of and conducted the experiments from which the main dataset used in my papers was collected.

It was a pleasure working with and getting to know so many brilliant scholars during my graduate career, specifically those members of VISLab; Dr. Rajkumar Theagarajan, who was my co-author on one paper and helped teach me about computer vision and deep learning; Dr. Federico Pala, who was my introduction to deep learning and was instrumental in the success of my thesis work; Dr. Asongu Tambo, Dr. Vincent On, Dr. Ninad Thakoor; Xiu Zhang, Padmaja Jonalagedda, Dr. Runze Li, Henry Liu, Ankhith Jain and Belinda Le. To those friends and collegues I met at UCR; Dr. James Stumpff, Dr. Troy Alva, Dr. Rachel Behar, and many others who helped me spiritually and professionally. Without you I would not have made it through this journey, thank you.

This work was graciously supported by the Tobacco Related Disease Research Program (TRDRP) Pre-doctoral Fellowship Award (DT27-0007) (www.trdrp.org) as well as the IGERT Fellowship in Video Bioinformatics (NSF DGE 0903667). These grants made possible the work comprising this thesis.

To my parents for their unconditional love and support. To my wife, who makes me want to be a better person. To my high school science teacher, the late Grant Glausser, who inspired me to be myself.

ABSTRACT OF THE DISSERTATION

Classification of Stem Cell Microscopy Images Using Deep Learning

by

Adam John Witmer

Doctor of Philosophy, Graduate Program in Bioengineering
University of California, Riverside, June 2024
Dr. Bir Bhanu, Chairperson

Stem cell biology has the potential to solve many problems related to disease modeling, and personalized regenerative medicine. Experimental research involving stem cells is often quantified via non-invasive microscopy imaging to observe morphological behavior and determine the underlying mechanisms of observed patterns. The image data collected from these experiments are very large and require extensive by-hand analysis by a skilled biologist to evaluate experimental outcomes. To this end, computer vision programs aimed at feature extraction and classification have become an indispensable tool to accelerate and standardize the analytical pipeline. Furthermore, deep learning has automated these programs to improve the accuracy and reliability of the results obtained from analysis.

There remain limitations imposed on deep learning that are a consequence of the unique circumstances under which the data are collected. For example, class imbalances, lack of data, ground-truth annotation, contiguous class boundaries, and multi-label images, to name a few. The goal of this thesis is to overcome these limitations through the novel use of biological features which inform and guide model design and implementation.

The major works comprising this dissertation involve deep neural network-based methods for stem cell microscopy image classification. Specifically, these methods address complex biological dataset problems including contrastive learning for improving discrimination of overlapping features, unsupervised generative adversarial networks for supplementation of limited datasets, and semi-supervised, pseudo-labeling to overcome costly manual annotations. Paramount to the success of these projects is the use of domain knowledge in the form of biological relationships between image classes, temporal and dynamic features, and microscopy scale-space, which are exploited to improve classification performance and are shown to be indispensable in designing efficient and effective deep learning models.

These novel video-bioinformatics methods are implemented to draw inferences of experimental endpoints from a dataset of microscopy images concerned with determining the effects of nicotine on a Huntington's disease iPSC model. Colony regions of interest are detected from raw, time-lapse microscopy images and sorted into four separate morphological classes that correspond to major cellular phenotypes. The proportions of these classes within the images over time provides insights into the growth and developmental changes that occur during these experiments. Deep learning is a powerful tool that automates the analytical pipeline and reveals key features of input images that can be used to model stem cell differentiation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Stem cells are a promising avenue for personalized medicine and disease modeling. Experimental models involving stem cells are often quantified using non-invasive cellular microscopy imaging to observe dynamic behavioral and developmental changes. One major hurdle in experimental analysis is the unstandardized and biased process of feature modeling, and classification. Traditional video-bioinformatics approaches for automating image analysis require substantial domain expertise for feature and classifier design. More recently, deep learning and artificial intelligence have simplified this process by combining these steps into a single algorithm that improves the accuracy and efficiency of these processes. However, data specific problems necessitate the design of novel deep learning models that incorporate domain knowledge and bio-inspired learning. To overcome these problems, the body of work proposed in this thesis introduces domain-inspired deep learning networks and computer vision algorithms for automated analysis of stem cell toxicology experiments. The outline of the thesis is described as follows:

Chapter 2 outlines related works to the proposed thesis, highlighting their advantages and disadvantages, and elaborating on two initial projects that were completed to address these limitations and served as a nexus for the dissertation work. The drawbacks of the straightforward implementation of deep neural networks for modeling of complex biological data are noted and used as inspiration for the body of the thesis work, which involve more sophisticated, domain-inspried implementations of deep learning to overcome issues involving dataset imbalances, feature entanglement, and expensive manual annotations.

Chapter 3 presents a method for addressing dataset imbalances using Generative Adversarial Networks (GAN) [1]. In this paper, GAN's are used to supplement a dataset of real microscopy images with generated image patches. This approach addresses the problems of imbalanced and limited biological datasets for training deep learning models. Temporal relationships between image classes are exploited to improve classification for multi-stage, hierarchical classification networks that filter images based on the downstream process of stem cell differentiation. The results of this work indicate that GAN generated images are a viable method for supplementing imbalanced datasets for neural network classification.

Chapter 4 details a project aimed at feature disentanglement using contrastive learning networks [2]. In this paper, contrastive triplet-networks are used to exploit the visual similarities between closely related morphological classes. A novel method for determining representative template images for model testing is presented that improves the classification accuracy of a hierarchical deep neural network. This approach highlights the importance of using domain knowledge for designing modeling architectures to discern be-

tween features of closely related morphological classes.

Chapter 5 introduces Iterative Pseudo Balancing, which overcomes the need for large, annotated datasets by using a teacher-student network to provide pseudo-labels to input image patches [3]. An pseudo-labeled dataset is balanced on the fly, and multi-scale inputs from multi-label images are incorporated to improve network learning. The combination of local and global features from image patches is shown to improve features extraction and the use of psuedo-labeling for training a neural network. In turn, this allows researchers to implement deep learning networks without having to manually annotated every data point, which is a time consuming and biased process.

These three works demonstrate the ability of deep learning models to accurately classify stem cell microscopy images and predict experimental endpoints using only morphological features. This non-invasive analytical approach allows for real time analysis of biological images and helps to streamline the analytical pipeline by simplifying and standardizing feature extraction and classification. The results of these works reveal several key insights into the biological effects of nicotine on Huntington's disease iPSC's, as well as having broader implications on personalized regenerative medicine and automated experimental analysis.

Chapter 2

Related Work

Traditional computer vision and machine learning algorithms have been successful in improving feature modeling and classification of biological images to overcome the limitations of by-hand image-data analysis, which is an inefficient, biased and tedious process. These methods involve determining a variety of morphological, texture or statistical features from a dataset of images, and training a machine learning classifier (e.g. decision tree, support vector machine, k-nearest neighbors, etc.) to determine and categorize important relationships between features. These models have simplified and standardized the image analysis process by removing sources of human error to automatically quantify experimental outcomes.

However, traditional computer vision methods require the determination of representative statistical features for each image class such that classification is based on the modeling of the most important features of each image. For example, Zahedi et. al. [4], determine the health status of stem cell colonies in time-lapse videos by classifying morpho-

logical and temporal features and correlating visual observations to experimental biomarkers. They determine 24 individual features including image pixel intensity, colony area, perimeter, radius, aspect ratio, and number of protrusions, as well as features related to mobility (i.e., centroid displacement) and growth (i.e., rate of change in area). They test exhaustive combinations of these features and train various machine learning classifiers to determine the most effective method for classification. They achieve 97% accuracy by training a support vector machine learning classifier to separate classes based on the number of protrusions and the minimum pixel intensity value.

While these results are promising for the application of stem cell microscopy image classification, there are some limitations to this method such as image field of view and complexity, and the aforementioned requirement of domain expertise in feature construction and classification. For example, the input images used in their study contain views of the entire cell colony within the image boundary, which is a result of image acquisition parameters, by which a single colony is placed at the center of the microscope at the beginning of the experiment and images are taken of the colony at the same position for every image. This allows for tracking of the centroid of the image, and for features such as shape and area to be calculated from the region of interest within the image. However, this ideal case is not always the situation when dealing with complex cellular experimentation, including large-scale experiments involving images under lower optical magnification, where several hundreds of cell colonies are observed at one time.

The requirement of domain expertise for feature selection is another bottleneck that limits the efficiency of experimental analysis using traditional computer vision methods.

For this, the researcher must determine relevant features from the dataset that can be extracted from images to classify data samples. This necessitates at strong understanding of what the data will look like at any given point in experimentation, and subsequent post-processing to extract these features from each image for training, testing, and validating a given machine learning model. Furthermore, the specific features or combination of features that provide the most relevant information for class discrimination must be determined empirically, which is a difficult and tedious process.

The first two projects completed in this dissertation aim to overcome the limitations of conventional computer vision methods by using deep neural networks to classify stem cell microsocopy images. Deep neural networks, which automate the feature extraction and classification processes into a single analytical pipeline. Convolutional neural networks (CNN) are deep neural networks that process images using the convolution matrix operation. The employ reinforcement learning to iteratively optimize feature maps comprised of millions of weighted parameters that capture important visual features of training images. Features are cascaded through many layers of these feature maps using a bio-inspired rectified linear activation unit that gates each layer, allowing only strong features to move to the next layer. In these primary works, several variations of CNN's have been designed to solve common dataset problems including the classification of cellular morphology and generation of synthetic images using generative adversarial networks.

The first work by Witmer et. al. [5], implements an automated computer vision pipeline to automatically quantify stem cell experiments. First, cell colonies are detected, segmented and cropped from large scale, raw microscope images containing hundreds of

colony ROI's. Then, deep neural networks are used to classify patches of stem cell colonies from large scale images. Two CNN configurations are shown to improve classification performance over standard, texture-based computer vision, machine learning methods. Finally, image patch predictions are overlaid on top of binary map areas to quantify and visualize cell colony phenotype over time. The growth rate curves of cell colony class-area reveal that nicotine does not induce neurogenesis, but also does not have any toxic affect on stem cell growth and development in a Huntington's disease stem cell model, and may increase the overall rate of growth of cell colonies over the course of the experimental time-frame.

This work demonstrates the power of deep learning to model complex cellular morphology while simplifying, standardizing and improving the feature extraction and classification pipeline for cellular microscopy experiments. However, there are some drawbacks to the straightforward implementation of CNN's in this work that affect model learning and performance. These include the effect of class imbalance on model learning, the requirement of extremely large training datasets, misclassifications caused by class overlap, and the scale of image patches, which are addressed in subsequent works presented in this thesis.

The second work by Witmer et. al., [6], overcomes problems associated with dataset limitations caused by the difficulty of gathering experimental data in a research setting. Deep neural networks require large datasets of millions of samples to learn an effective mapping of input data. Experimental research involving stem cell microscopy imaging is a difficult and time-consuming processes that results in limited datasets that often are insufficient to efficiently train neural networks. The method proposed in this work overcomes this limitation by generating new image patches using Generative Adversarial Networks (GAN)

that are used to supplement the dataset without running more experiments, saving time and money in regards to experimental data collection.

GANs are an unsupervised deep learning network that use features learned from input images to generate new image samples from Gaussian noise input. The model is comprised of two networks, the generator that takes a noise vector as input and outputs a random image; and the discriminator, which takes an image as input and outputs a prediction regarding the realness of the image. Over many iterations, the GAN learns to generate more realistic images as the model learns a better representation of the real dataset. In this work, the trained GAN model is used to generate images that are be added to the real dataset for training a CNN to perform four-class classification. The models supplemented with GAN generated images out-perform standard CNN models in terms of classification accuracy, which is due to the injection of image features during training. Some limitations of this work include class imbalances in the dataset, limited training configurations, and generated image quality control, which are addressed the work described in Chapter 3.

The neural network architectures employed in these initial works highlight the limitations of a straightforward implementation of state-of-the-art deep neural network methods for complex biological datasets. Several factors surrounding biological data collection and curation necessitate domain specific design of deep learning architectures and image processing pipelines. For example, the presence of class overlap in multi-label images, class imbalances caused by downstream differentiation, and issues arising from the difficulty in collecting experimental data to train deep learning models. To this end, the proposed thesis overcomes these issues by implementing domain-specific, bio-inspired deep learning models

of stem cell growth and differentiation that take into account the experimental and data collection procedures and underlying biology that lead to unique patterns observed in the image data. The works described in the following chapters represent novel implementations of deep neural networks for stem cell experimentation with implications for a wide variety of applications. The results presented here demonstrate the power of deep learning to determine underlying biological mechanisms of cellular behavior using morphological observation of non-invasive microscopy imaging.

Chapter 3

Generative Adversarial Networks for Morphological-Temporal Classification of Stem Cell Images

3.1 Abstract

Frequently, neural network training involving biological images suffers from a lack of data, resulting in inefficient network learning. This issue stems from limitations in terms of time, resources, and difficulty in cellular experimentation and data collection. For example, when performing experimental analysis, it may be necessary for the researcher to use most of their data for testing, as opposed to model training. Therefore, the goal of this paper is to perform dataset augmentation using Generative Adversarial Networks (GAN) to increase the classification accuracy of Deep Convolutional Neural Networks (CNN) trained

on induced pluripotent stem cell microscopy images. The main challenges are: 1. modeling complex data using GAN and 2. training neural networks on augmented datasets that contain generated data. To address these challenges, a temporally constrained, hierarchical classification scheme that exploits domain knowledge is employed for model learning. First, image patches of cell colonies from gray-scale microscopy images are generated using GAN, and then these images are added to the real dataset and used to address class imbalances at multiple stages of training. Overall, a 2% increase in both true positive rate and F1-score is observed using this method as compared to a straightforward, imbalanced classification network, with some greater improvements on a class-wise basis. This work demonstrates that synergistic model design involving domain knowledge is key for biological image analysis and improves model learning in high-throughput scenarios.

3.2 Introduction

Stem cells are unspecialized cells that are used as a model for early-stage growth. They recapitulate biological characteristics of embryonic development, most importantly pluripotency, or the lack of specified cellular purpose [7]. Deviations from this pluripotent state are an indication of differentiation, or phenotypic lineage commitment, and have implications on the health and developmental status of cells and cellular colonies [8].

The normal growth and downstream differentiation cycles of pluripotent stem cells are highly coordinated and delicate processes. Much work has been performed to delineate and manipulate these molecular changes in-vitro in order to better understand the mechanisms by which they occur [9, 10, 11]. For example, adult cells have been turned back into

stem cells in-vitro (induced pluripotent stem cells, iPSC's), and normal stem cell differentiation has been modeled using Markovian stochastic methods [12, 13, 14].

These and other studies have determined that stem cells transition from the pluripotent state to the differentiated state via an intermediate progenitor and that many unobservable sub-states exist within these larger, observable phenotypes. These phenotypes present themselves with distinct morphological structure and can be observed via light microscopy as cellular colonies with unique gray-level texture patterns. Depending on the duration of in-vitro differentiation, varying proportions of cellular colonies at given stages are observable at different points in time.

For example, in the beginning of the process, there are more of the early-stage cell class, while in the middle all three stages can be observed, and there may be some of the late-stage, and more of the intermediate stage. Towards the end of this cycle, given a high yield, there should be more of the fully differentiated stage than the earlier two classes. Figure 3.1 displays the nature of the multi-class cell colonies with contiguous cell boundaries that result in the four morphological classes used in this paper (Debris, Dense, Differentiated, and Spread) as described in Table 3.1. The spatiotemporal way data are collected over the course of this process provides snapshots of each of these stages. The normal differentiation process is subject to both internal cues and external factors, and the balance of these signals can influence cellular fate. Stem cells are particularly susceptible to external perturbations, and early molecular changes such as DNA mutations can have long term effects on cellular and organism health.

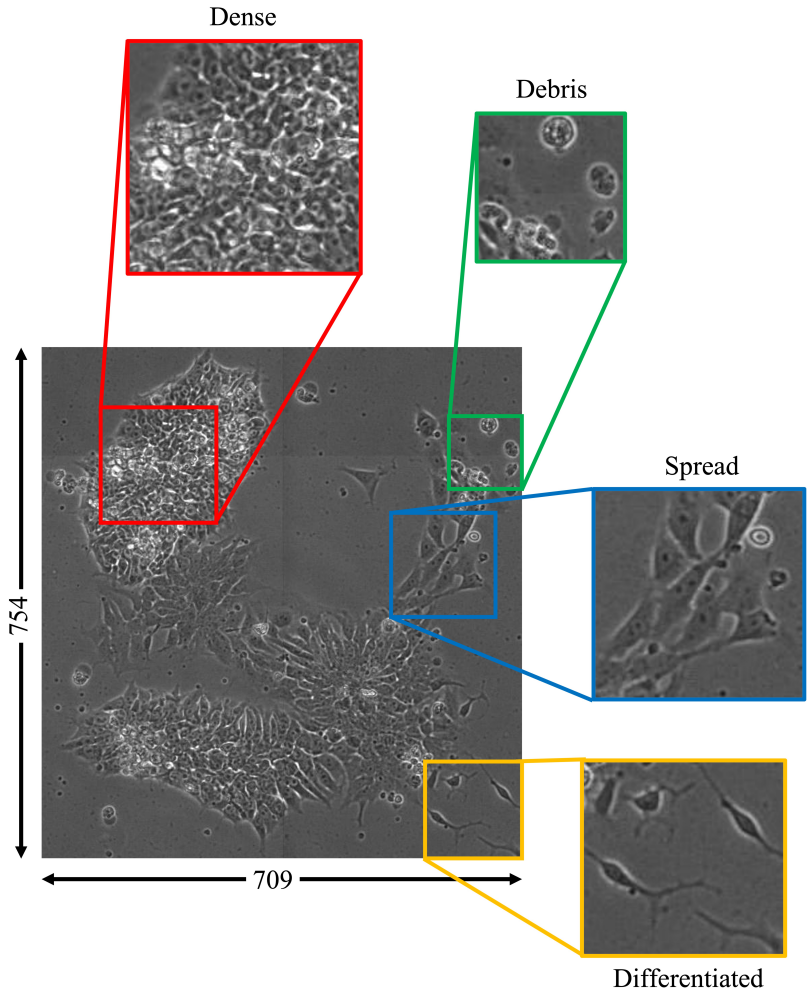

Figure 3.1: Image examples for four morphological classes observable in a single cell colony (Debris: green; Dense: red; Spread: blue; Differentiated: yellow). Throughout the differentiation process various proportions of each class can be found in cell colonies with contiguous cell boundaries. Classification of these multi-class images can be performed using image patches.

3.2.1 Developmental Toxicology

Developmental toxicology is the study of the effects of environmental factors on pre-natal growth and development [15]. In-vitro studies aimed at observing the developmental effects of exposure to tobacco chemicals on stem cells have been influential in the formation of United States Food and Drug Administration (FDA) policies for harm reduc-

Class	Morphological Description	Implication
Debris	Individual cells or aggregates cells showing circular morphology with high intensity white 'halo' marking distinct boundaries	Distressed, dead (apoptotic/necrotic) cells that float on top of colony indicating negative response to experimental conditions
Dense	Homogeneous aggregates of small cells with indiscernible cell boundaries, no clear nucleus	Induced pluripotent stem cell colonies that maintain undifferentiated status under current conditions
Spread	Homogeneous aggregates of large cells with discernible cell boundaries, clear nuclei, large protrusions	Down stream lineage intermediates or progenitor cells
Differentiated	Individual cells or spaced out aggregates of cells with distinct, dark cell bodies, high-intensity white boundaries, and dark axon like protrusions.	Differentiated neurons or neuron-like downstream lineages

Table 3.1: Morphological Class Descriptions and Corresponding Biological Implications

tion and public health. However, the pace of stem cell research is often limited by the tedious, and time-consuming process of image data analysis. In addition, the desired scale of experiments involving the testing of multiple chemical compounds at various concentrations, across multiple cell types, in a high-throughput manner, results in data that are often impossible to analyze by hand. Recently, computational analysis using video-bioinformatics has become an invaluable tool for researchers towards reducing human error and increasing analytical throughput.

3.2.2 Video-bioinformatics and Machine Learning

According to Bhanu et al. [16], Video-bioinformatics (VBI) is "the automated processing, analysis, understanding, data mining, visualization, query-based retrieval/storage

of biological spatiotemporal events/data and knowledge extracted from videos obtained with spatial resolution varying from nanometer to meter of scale and temporal resolution varying from seconds to days and months." Many aspects of experimentation can be quantified by observing cellular behavior in images and videos in a non-invasive manner (i.e., without killing or otherwise perturbing live cells). Light microscopy is often used to observe dynamic colony behavior by collecting time-lapse images during experimentation. Data collection is sometimes automated using an incubator-microscope unit such as the Nikon Biostation CT to accumulate temporal image data [17]. These units are programmed to collect whole-dish images, or perform single colony tracking at desired time intervals, for extended periods of time, helping to standardize data collection in a high-throughput manner.

A bottleneck arises in the analysis of resulting image data, normally processed by hand over the course of many weeks, or with the aid of open source software such as ImageJ and CLQuant [18, 19]. These rudimentary programs require users to sift through their dataset, image by image, selecting, outlining/tracing, and visualizing images or creating image processing pipelines designed to generalize across the dataset. These algorithms are useful tools for segmenting and measuring objects in an image but require substantial expertise and user input, which comes with the possibility of increased error due to non-standardized bias.

In the past decade, many advances have been made to remove user error and bias by automating the image analysis process. The utility of VBI programs has expanded to include quantification and classification of results. In general, these image-processing programs are concerned with leveraging unique characteristics (i.e., features) of images, at both the pixel

and image level, to accomplish a desired task, such as classification. Some examples of features include global static information such as colony size, shape, and morphology, as well as temporal information including motility, growth rate, and observed behavioral status (e.g., differentiation or death) [4]; local features include information found within patterns, such as texture, contrast, intensity, and color. Programs that exploit these features are useful to researchers for removing sources of human error by combining, standardizing, and automating the feature extraction and quantification/classification processes. For example, Guan et al. use a Gaussian mixture model to segment stem cells from image background in static microscopy images [20].

Many of these programs also employ machine-learning algorithms such as clustering, decision trees, or support vector machines to improve classification. One such program, called StemcellQC, analyzes time-lapse microscopy videos using pre-determined, hand-crafted morphological features of stem cell colonies. This program takes input from the user via a graphical user interface (GUI) in terms of setup and desired output, and automatically analyzes and plots outputs for the user to view [4]. Global features such as colony area, aspect ratio, and motility are combined with local features, including gray level intensity, to classify individual colonies by health status (healthy, unhealthy, dying) using standard machine learning algorithms to determine the effects of toxic chemicals on cellular behavior.

Another program, Pluri-IQ uses a supervised random forest classifier to distinguish between cell colonies at different stages of growth from pluripotent to differentiated in dense, fluorescent microscopy images [21]. While these software improve the efficiency of

analysis through standardization, they still require user interaction, rely on the researcher to pre-determine features based on prior knowledge of colony behavior, or exploit some previously observed pattern. More recently, deep learning has revolutionized image analysis by automating both the feature extraction and classification processes to remove sources of human error and bias. The following section discusses deep learning approaches for biological image analysis.

3.2.3 Deep Learning Approaches

Deep learning (DL) programs help to overcome the drawbacks of data analysis by combining feature extraction and classification into a single model. DL models do this by determining mathematical features of images that can be used to categorize input data. These features consider all aspects of an input image and are iteratively refined with respect to a desired output using a gradient descent optimization algorithm. Moreover, deep learning extends these algorithms to image processing and analysis via Deep Convolutional Neural Networks (CNN). The term 'deep' comes from the layered architecture of the CNN; 'neural network' comes from the weighted connections between a pair of layers, and a bio-inspired gated activation operation, the 'Rectified Linear Unit' (ReLU) is like the all-or-nothing activation response of neurons to an input signal [22]:

$$f_{\text{node}} = \sum_{i=1}^{n} (w_{\text{i}} x_{\text{i}}) + b \tag{3.1}$$

$$ReLU_{\text{y}} = \begin{cases} y = 0 & y \leq 0 \\ y = y & y > 0 \end{cases} \tag{3.2}$$

These algorithms sequentially multiply input images by the layered weights as shown in Equation (3.1). At each node, the output is f_{node}, x_{i} is the input value from the previous layer, w_{i} is the weight value of the layer, and b is an additional bias term. Positive valued outputs are fed forward to the next layer through a piecewise, ReLU activation function (Equation (3.2)), where y is the feature map from the previous layer. The output of these operations at the end of the network is a unique numerical signature that is used to determine the class of the image.

During training, predictions of the network are used to update the parameters (weights) of the model with respect to a given ground-truth. While DL was initially employed to classify extremely large, real world datasets such as ImageNet [23], it has been used recently to improve the accuracy and efficiency of analysis for bio-medical applications including microscopy [24, 25, 26, 27], high-throughput methods [28], MRI [29], histopathology [30], and stem cell microscopy imaging [31]. For example, in our previous work, the patch-based classification of multi-label colony images was addressed in [5]. However, the aforementioned caveat of dataset size, as well as the large size (224 x 224) of image patches used in this work were noted as drawbacks to this method.

The high-parameter nature of these networks requires training them with extremely large datasets to avoid overfitting, which is the case where the model learns to classify the training images perfectly, instead of learning features that generalize well across the testing dataset. There is commonly a positive correlation between dataset size and network accuracy, because the more sample images the network sees during training, the more precisely it models the data and, consequently, the more information it uses to make decisions during testing. Unfortunately, the nature of biological experimentation frequently limits the size of the dataset, based on time, resources, and general difficulty in performing experiments.

Therefore, there is a need to increase the size of biological datasets by supplementing images with similar and visually relevant data, without having to perform new experiments to collect more images. Generative Adversarial Networks (GAN) provide a unique opportunity to generate new images that are representative of the real data. This paper is aimed at performing data augmentation by supplementing a minimal biological dataset via image generation using GAN [32]. GAN are a subset of DL networks that combine two opposing networks that take a noise vector as input and produce an image based on the features that they learn from the real dataset. More information on these complex networks is provided below.

3.2.4 Generative Adversarial Learning

GANs are DL models that combine two networks, a Generator (G), and a Discriminator (D), that play a min-max learning game to model input data (Equation (3.3)). G takes as input a vector, z, of numbers sampled from a Gaussian distribution, and performs

up-sampling convolutions to produce and n×n size image, *G(z)*. *D* alternately takes as input either real or generated images and performs down-convolutions to produce a realness score that is used to determine if the image is real or fake. The goal of G is to generate images that fool D into thinking that they are real (i.e. minimize the probability that G comes from the fake distribution, p_z), and the goal of D is to maximize the log-likelihood probability that a given image comes from the real distribution $p_{data(x)}$, where E is the expected value operation:

$$\min_G \max_D V(D,G) = E_{\mathrm{x}\sim \mathrm{p_{data(x)}}}[\log\ D(x)] + E_{\mathrm{z}\sim \mathrm{p_z(z)}}[\log(1 - D(G(z)))] \tag{3.3}$$

GANs learn a representation of input features in an unsupervised manner, which can then be used for additional downstream learning tasks such as feature extraction and classification [33]. Convolutional GANs were originally designed to generate images from extremely large, open-source datasets, such as natural images (Imagenet [22], CIFAR [34]), faces, and numbers (Mnist, [35]). More recently, this work has been expanded into unique datasets including those from medicine and biological experimentation. Efforts to address the challenges of modeling a unique cellular microscopy dataset using GAN, and the use of generated data for dataset augmentation, are the subject of this work. Related works involving GAN are presented in the following section.

3.3 Related Works

Recently, GANs have been used for biological image generation tasks involving cellular microscopy and medical image datasets [36, 37]. For the purposes of this paper, cellular microscopy images are considered distinct from medical images (e.g., x-ray, CT, MRI) in that they deal with objects on a micrometer scale and are usually obtained from cellular experimentation involving microscopy. Factors including scale and cellular morphology increase the visual complexity of data and, in turn, that of modeling/analysis. There are many ways to implement GANs for learning applications using cellular image data with the goal of gathering useful features in an unsupervised manner. Some examples of cellular microscopy applications of GAN are given below.

One indirect method of using GAN features is through transfer learning, in which the information learned from GAN, in the form of network parameters, are leveraged by separate models to improve network performance when using few labeled data. For example, Majurski et al., [38] use features from GAN trained on fluorescent stem cell microscopy images to perform cellular segmentation with a U-net style pixelwise classification network. Wang et al., [39] use GAN discriminator features to fine tune a classifier on minimal labeled dataset for detection of 'rosette' formation in early stage C.elegans embryos. While transfer learning using GAN can indirectly inform downstream model learning, GAN can also be used to directly produce image outputs for observation of the learned representation.

An example of this is the task of image-to-image translation, in which the original style or modality of an image is transformed to another of desired nature. Rivenson et al., [40], use a VAE-GAN style network, in which an encoder-decoder-discriminator network

is used to translate images acquired using a non-invasive phase contrast microscopy, into colorized histology images. This 'digital staining' technique is used to circumvent the difficult process of histology staining, in favor of non-invasive microscopy techniques. Lee et al., [41] perform 3D fluorescent microscopy deconvolution using a specialized CycleGAN framework. This network performs image-to-image style transfer using multiple GANs to learn a mapping between unpaired images of different styles, such that there is no need for a direct ground-truth comparison during training. They use this framework to sharpen noisy images for improved segmentation in 3D volumes of rat kidney sections. Bailo, et al., [42] generate images of red blood cell smears to perform dataset augmentation for segmentation and detection tasks. They train a sophisticated image generator (pix2pix) to perform image-to-image translation from segmentation masks of real images and subsequently generate new images from synthesized segmentation masks. These works utilize GAN to change images from one style or modality to another, but do not directly employ GAN for image generation from a latent space.

More straight forward implementations of GAN involve the use of networks to directly generate images from a learned feature representation using only a latent variable as input. Goldsborough et al., [43] generate single cell fluorescent images in three color channels using various GAN models and perform image interpolation before using GAN features for transfer learning, observing increased classifier performance. Pandhe et al., [44] combine a GAN image representation with autoregressive motion synthesis to accurately recapitulate neutrophil behavior and observe patterns of organelle function. Theagarajan et al., [45] generate single cell images of human embryonic stem cells across 5 health-related classes

using multiple networks. They use an ensemble of GAN networks to generate thousands of stem cell images for dataset augmentation and find increases in evaluation metrics for the number of added images. Osokin et al., [46] use a "separable generator" to generate two color channels of a multi-channel fluorescent images. In all these cases the generated images are of whole cells, where the entire cell body is within the field of view. Images like these are generally less difficult to model than the more detailed, varied, and fine-grained texture patterns observed in the image patches generated in this work, because the model can learn the relationship between the background and foreground.

The work of Devan et. al., [47] provides another example of a GAN implementation for limited biological dataset augmentation. The authors of this work use GAN to increase the size of a transmission electron microscope image dataset. They perform automatic detection of cytoplasmic capsids using Region Based CNN (R-CNN) [48] with an augmented dataset, and show an improvement of their results against the standard dataset configuration. This method employs SinGAN [49] to generate alternative versions of real images using a pyramidal GAN network. Unlike this work, the proposed method seeks to use GAN to generate completely new image patches to be added to the dataset, instead of different versions of real images. Furthermore, the proposed method also takes into account class relationships and dataset imbalances caused by experimentation.

Similarly, Dimitrakopoulos et. al., [50] perform GAN based dataset augmentation for open-source medical image datasets. They propose a GAN model, Ising-ResGAN, that uses Markov random field constraints to perform image smoothing. They use generated images to improve the results of a U-net segmentation task for various publicly available

datasets. They generate images with a large field of view (256×256) and do not include domain specific knowledge, where as the proposed method generates sub-colony image patches and incorporates a learning scheme based on stem cell differentiation.

3.3.1 Contributions of this Paper

While the above related work represents useful applications of GAN to biological datasets, none of them addresses the issue of temporally constrained differentiation. The previously mentioned works involve the modeling of static images with varying levels of image complexity in terms of cellular structure and colony density, texture, and overall variation across datasets. These works do not impose biological constraints or exploit domain knowledge. In this paper, a mathematical model of stem cell differentiation, involving Markov-chain stochastic processes, is used to inform model training.

The focus of the proposed approach is on the generation of small image patches containing fine grained texture features and high variation, across four classes (Dense, Spread, Differentiated, and Debris) using GAN. This paper expands significantly on previous work in [6] by testing multiple model configurations/architectures and introducing a new GAN training/quality control scheme that utilizes image entropy distributions for improved training. These aspects of the proposed model make it novel in comparison to the previous work. The contributions of this paper are as follows:

1. Models complex, varied, and highly textured image patches using GAN

2. Incorporates domain knowledge in the form of temporal constraints on model learning as well as bio-inspired algorithm design

3. Introduces an image-entropy based metric for model training, image post processing and quality control

4. Explores dataset augmentation as a viable means for improving network performance for tasks involving patch-based classification

The data used in this work present unique challenges in terms of image generation and classification. Specifics of the proposed method to address these challenges are discussed in the following section.

3.4 Materials and Methods

3.4.1 Technical Approach

Figure 4.2 describes the overall approach for colony detection and image pre-processing used in this work. The pre-processing step is performed to reduce the amount of flat background in images, which contains no class-relevant information. The large size images from this dataset are too computationally expensive to be processed as whole images, and relevant colony areas make up only a portion of the original stitched microscope image. Cell colonies are detected using a morphological segmentation algorithm (sequential operations: 3×3 Gaussian blur, entropy filtering (disk filter,size 3), morphological opening (disk filter size 3), binarization via Otsu thresholding, hole filling, small object removal ¡2000 pixels). Detected colonies are cropped out to amass a dataset of colony image ROI's. After this, random patches of cropped images are used to train separate GAN models for each class.

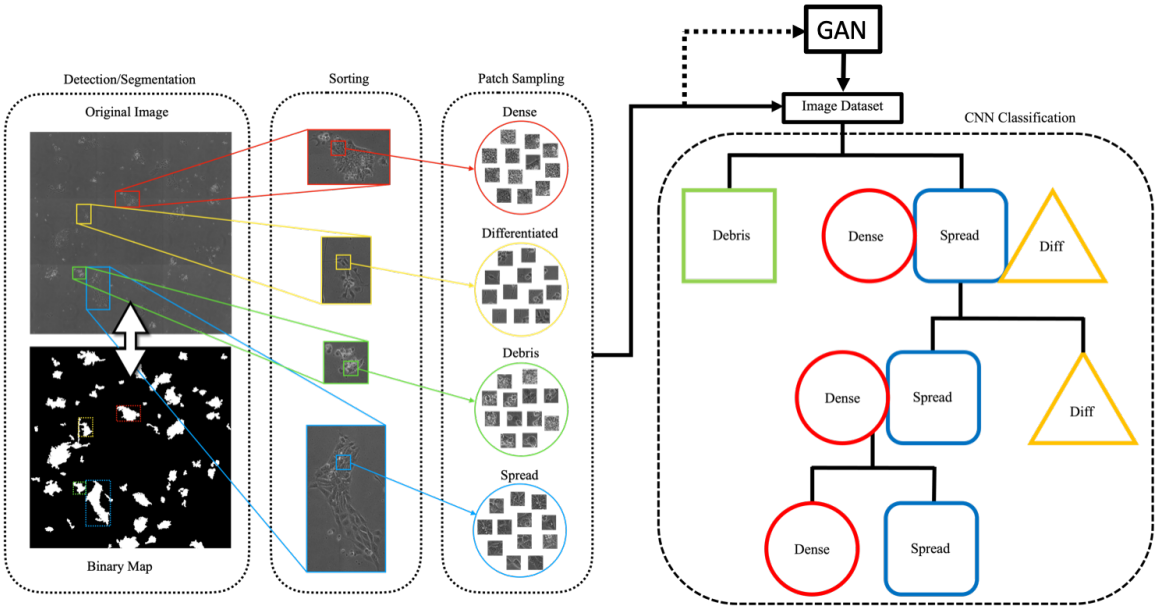

Figure 3.2: Data pre-processing and classification schematic. The binary map of colony locations is used to segment colonies from the original image, which are then sorted by hand during ground-truth generation. Patches from the resulting dataset are used to train the GAN. Generated images are added to balance the dataset for the Temporal CNN Classification scheme (right), during which images are sorted into their individual classes through multiple hierarchical stages.

The proposed method employs GAN to model four separate data classes (Dense, Spread, Debris, Differentiated; see Table 3.1). These classes are determined by the morphological appearance of cell colonies in static images, and correspond to cellular phenotypes, or specific cell types, based on prior biological knowledge. The use of multiple GAN networks (one for each class) in this work has several advantages including: 1. negating the effects of class imbalances; 2. improved training because of modeling a unimodal distribution of class features; 3. allowing for the specific tailoring of each GAN for a single class (i.e., number of training epochs for convergence); 4. feature-disentanglement via extraction of class-wise information, which includes the entropy loss calculated during training.

3.4.2 GAN Architectures

In this work, several Generative Adversarial Network (GAN) architectures and loss functions are tested to determine the most advantageous configuration for the specific task and dataset. Given that GAN training is notoriously unstable, a Deep Convolutional GAN (dcGAN) [33] architecture is adopted as a base model. The GAN Generator, G, takes as input a 100-dimensional Gaussian noise vector, and outputs a 64×64 grayscale image. That image is then fed alternately with real images to the discriminator, D. D then outputs an adversarial (real/fake) score for the image, which is the binary cross-entropy criterion, where L is the loss value, x is the network output with respect to the input c, and is computed across all samples, j Equation (3.4).

The standard loss function for the generator is the adversarial loss with respect to the generated data distribution, P_{z}, (Equation (3.5)). The aggregate loss function for the discriminator is the average of the adversarial losses for both the generated, P_{z}, and real image samples, P_{data}, (Equation (3.6)). GAN training uses the Adam optimization algorithm along with the parameters provided in Table 3.3. These learning values are determined empirically using network optimization, and allow for stable, efficient training of the GAN.

$$\mathcal{L}(x, c) = -x(c) + \log(\sum_{j} e^{x_j}) \tag{3.4}$$

$$\mathcal{L}_{\mathrm{dcGAN_{Gen}}} = \mathcal{L}_{\mathrm{Adversarial_{P_z}}} \tag{3.5}$$

$$\mathcal{L}_{\mathrm{dcGAN_{Dis}}} = \mathcal{L}_{\mathrm{Adversarial_{P_z}}} + \mathcal{L}_{\mathrm{Adversarial_{P_{data}}}} \tag{3.6}$$

Generator		
Module	Size	Maps
Linear	1×100/16	1/512
Up	16/36	512/512
C2d	36/36	512/512
Up	36/64	512/512
C2d	64/64	512/256
C2d	64/64	256/1
Tanh(·)	64/64	1/1

Discriminator		
Module	Size	Maps
C2d	64/32	1/64
C2d	32/16	64/128
C2d	16/8	128/256
C2d	8/4	256/512
FC	8192	512
Sig(·)	1/1	-/-

Table 3.2: GAN Network Architecture: The input to the GAN generator (a) is a latent vector of length 100 multiplied which is processed through multiple convolutional (C2d) and up-sampling (Up) layers. The output of the generator is a hyperbolic tangent (Tanh). The output of the discriminator (b) goes to a fully connected (FC) layer followed by a Sigmoid function (Sig).

Training Hyper-parameters	
Parameter	Value
Learning Rate - Adam	0.002
β_1 - Adam	0.5
β_2 - Adam	0.999
Max feature maps - Discriminator	512
Max feature maps - Generator	512

Table 3.3: GAN Training Hyper-parameters were empirically determined to optimize network training efficiency.

Several other GAN models and loss functions are compared to this baseline to determine the effect of GAN configuration on generated image quality. These architectures include the Wasserstein GAN (wGAN) [51], Auxiliary GAN (auxGAN) [52], and Metropolis-Hastings GAN (mhGAN) [53]. Each of these networks use the dcGAN as a framework on which to build specific training/learning techniques and loss functions. Brief overviews of each configuration are outlined below.

Wasserstein GAN

The Wasserstein GAN (wGAN) is a network configuration that tries to solve the problem of "vanishing gradients" in normal GAN applications that lead to the phenomenon of mode collapse in image generation. Mode collapse is when the generator fails to model all the variability in the input dataset, and instead learns to output images that contain only a small subset of input features. This is often due to the discriminator learning a very good mapping between real and fake images, which prevents the generator from training efficiently. wGAN attempts to control the weights of the discriminator by restricting, or "clipping", the highest and lowest weights within the discriminator feature maps to allow the generator to learn a more sufficient mapping of the dataset distribution during training.

Auxiliary GAN

The auxiliary GAN is a dcGAN network that uses image labels as a condition for network training. The image labels are provided to the generator network as a latent embedding, which gives the network prior information about image class. The generated images are then passed to the discriminator, which outputs both an adversarial score, as well as an auxiliary classification score in the form of Cross-Entropy Loss criterion (Softmax function + Negative Log Likelihood). The loss function for this network then becomes the aggregate of the adversarial and auxiliary losses for both the generator and discriminator, as shown below in Equations (3.7) and (3.8):

$$\mathcal{L}_{\mathrm{auxGAN_{Gen}}} = \mathcal{L}_{\mathrm{Adversarial_{P_z}}} + \mathcal{L}_{\mathrm{Auxiliary_{P_z}}} \tag{3.7}$$

$$\mathcal{L}_{\text{auxGAN}_{\text{Dis}}} = \mathcal{L}_{\text{Adversarial}_{\text{P}_\text{z}}} + \mathcal{L}_{\text{Adversarial}_{\text{P}_\text{data}}} + \mathcal{L}_{\text{Auxiliary}_{\text{P}_\text{z}}} + \mathcal{L}_{\text{Auxiliary}_{\text{P}_\text{data}}} \tag{3.8}$$

The advantages of the auxGAN configuration are that it allows multiple image classes to be produced by the same generator. However, the network has the more difficult task of modeling a multi-modal distribution, which could affect individual class-wise image quality.

Metropolis-Hastings GAN

The Metropolis-Hastings Generative Adversarial Network (mhGAN) is a dcGAN implementation that uses Markov-Chain Monte Carlo sampling to try to improve image generation. mhGAN attempts to find a more accurate image representation for the generator by using the discriminator to guide image selection via the Metropolis-Hastings algorithm. Equation (3.9) illustrates how the model is able to learn the relationship between the data distributions, $p_{\text{D}}(x)$ and $p_{\text{G}}(x)$ using the output of the discriminator, $D(x)$. The discriminator is provided multiple real and generated samples and is tasked with determining the most relevant real images with which to train the generator, based on its decision function:

$$\frac{p_{\text{D}}(x)}{p_{\text{G}}(x)} = \frac{D(x)}{1 - D(x)} \tag{3.9}$$

An Auxiliary implementation of the mhGAN is also trained here to test the ability of the Metropolis-Hastings algorithm to model the multi-modal distribution. All GAN networks are trained to convergence using the Adam optimizer, and training is monitored

Image Class	Overlap Percentage - Mean (std.)
Debris	0.6182 (0.0026)
Dense	0.7066 (0.0033)
Diff	0.3936 (0.0018)
Spread	0.3999 (0.0011)

Table 3.4: Overlap percentage for Image Entropy Histograms across 5 trials, for which the entropy values of 50,000 random real and generated image patches each are plotted and the overlap is calculated. Variation in values is caused by randomly generated image patches that contain variability within the image. These values can be used to determine how well the generator has been able to model image features, and can be correlated with the performance of down stream dataset augmentation tasks.

via loss function and generated image appearance. A table of network hyper parameters for training is provided in Table 3.3, where all values were empirically determined for optimization. Networks are trained using NVIDIA GeForce GTX 1080ti GPU's and programmed using the Pytorch deep learning library [54]. This model also utilizes the advice of GAN Hacks for design and implementation (https://github.com/soumith/ganhacks). The effectiveness of the various GAN methods in generating realistic images is assessed using multiple standard image quality metrics, as well as a novel metric introduced in the following section.

3.4.3 Assessing Generated Image Quality

In this work, several standardized methods of assessing generated image quality are used to compare all these implementations. Quality metrics such as Inception Score [55] and Frécet Inception Distance [56] are used, as well as a novel image entropy-based technique that is introduced in this paper, and described in the following section.

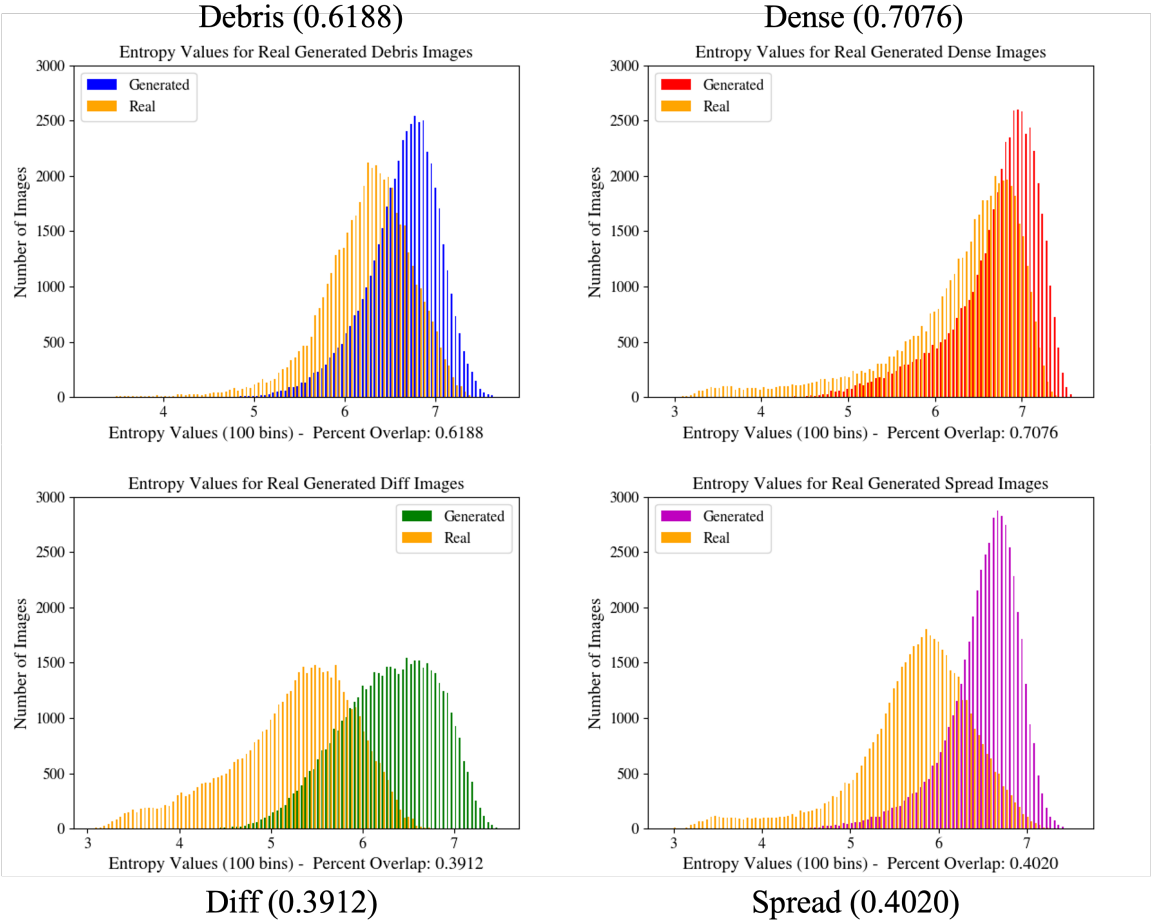

Figure 3.3: Image Entropy Distribution Histograms for GAN Configurations. These graphs provide a quantitative measure of the overall generated image distribution in relation to the real image distribution and are used during GAN training to improve network learning. Values in parentheses indicate the percent overlap of the two graphs shown in the figure.

Image Entropy Distribution

The image generation scheme proposed in this work results in image patches that contain various proportions of foreground/background area where foreground textures are representative of sub-colony cellular morphology. The random nature of image patch sampling allows for the network to learn both colony body and boundary areas, which are equally important to the overall classification task when trying to encompass the whole

colony area. One method of measuring the accuracy of the generated image distribution is by using image entropy. For this, image entropy is calculated as the Shannon Entropy (Equation (3.10)) of the individual generated image patches:

$$H = -\sum P_{\text{hist}} * \log_2 P_{\text{hist}} \tag{3.10}$$

During GAN training, image entropy is calculated for mini batches of real and fake images, and the normalized image-entropy probability distributions are used to find an image entropy loss parameter, L_H. Comparison of distributions is performed using the mean squared error (MSE, Equation (3.11)), evaluated over n samples using the squared difference between measurements Y and $\hat{Y}$ and then added to the aggregate loss function of the discriminator:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.11}$$

Trained GANs are used to generate image patches which are added to the real dataset at various percentages and subsequently used to train various CNN models to perform image classification across the four classes. The effect of this added value is observed by plotting the image entropy values of 50,000 random real and generated image patches (Figure 3.3). The percent overlap between the graphs is calculated by summing the smallest overlapping values across all the bins, and dividing by the total number of calculated values. Both the generated and real images have entropy values within the range of 3-8, but the generated images tend to be skewed towards the upper range of this distribution. Higher

image entropy is an indication of higher variability in the images and can be interpreted as images with more visual information.

Table 3.4 shows the average class-wise image entropy histogram overlap percentages for 5 trials of entropy histogram calculations. One trial consists of generating histograms using 50,000 random real and generated image patches each. Random image generation causes variation within these calculations, and is indicative of variability of generated images. These values measure the ability of the generator to model relevant class information, and can also be used to inform model learning.

It can reasonably be assumed that the greater the overlap between the real and generated image entropy distributions, the more accurately the generator is able to model the real image patches. These graphs also allow for the visualization of the image distributions in terms of the entropy values and are useful in determining the variability of features that are modeled by the generator. From Table 3.4, the classes with the lowest overlap values are the differentiated and spread classes. For the differentiated class, this may be due to the relatively small number of images available to the generator for modeling, or the difficulty in modeling features of the specific class. For the spread class, it may be due to the generator learning the very high-entropy features of this relatively large class, whereas the real image distribution displays a wider entropy curve.

In addition to calculating a training loss value, this entropy metric is also useful as a measure of the accuracy of the generated image distribution. These values can be correlated to the performance of the downstream data augmentation tasks as well as to the effect of changing the loss function of the discriminator using this entropy criterion

as discussed in Section 3.5 of this paper. The data generated by GAN are then used to augment the real datasets for training classification CNNs with temporally constrained configurations, as described below.

3.4.4 CNN Training Configurations

The main objective of this paper is to improve CNN performance for data limited settings involving biological images. To achieve this goal, the approach used is to augment the real dataset using GAN generated images as described above. However, this task is not as straightforward as it seems. There are many ways in which generated data augmentation can be effective for network training. Two common biological dataset issues are addressed here, namely data imbalances, and limited datasets.

Temporal Classification

The dataset configurations employing generated image augmentation are used in conjunction with temporally constrained, hierarchical classification CNNs. Temporal constraints are imposed according to the in-vitro differentiation process as modeled by Stumpf, et al. [13]. During this procession, which has been shown to recapitulate in-vivo differentiation, cells undergo downstream lineage changes that can be separated into three major categories: Embryonic-like Stem Cells (ESC), Intermediate progenitors, and Differentiated Neurons. Similarly, there are three classes of viable cell colonies that compose the dataset used in this work: Dense (ESC), Spread (progenitors), and Differentiated (neuron-like formations).

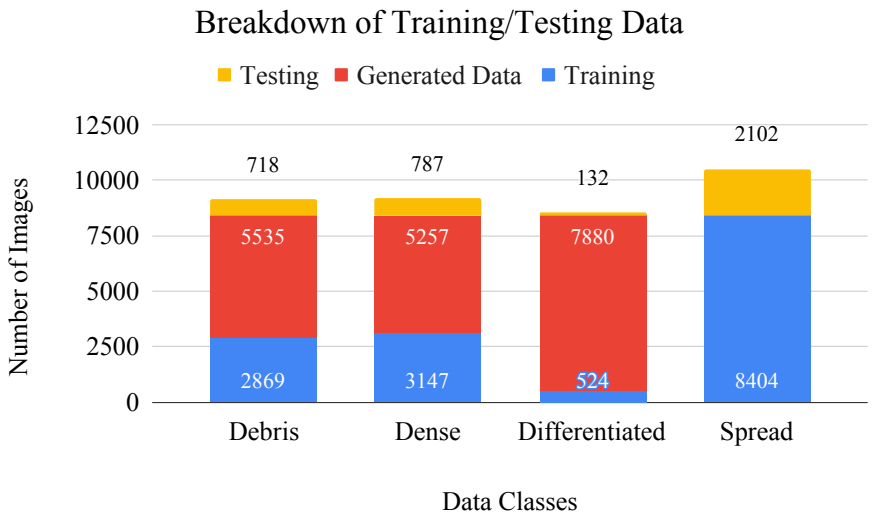

Figure 3.4: Bar graph of data breakdown including values for Training/Testing (Blue/Yellow) Split. Generated images (red) are added to the dataset to make up for class imbalances during CNN training.

Additionally, there are colonies of non-viable cells that are known as Debris, that represent dead or unhealthy cells and exist independently as well as within viable colonies. These areas are important because they characterize the adverse effects of toxic exposure and provide insight about the health status of cell colonies. Therefore, the hierarchical CNN classification system is set up as a series of two-class networks that combine the various class-stages of growth and differentiation as described above. Figure 3.4 provides a visual reference for data imbalances, as well as the number of images provided for the train:test split. The number of added images in various configurations is shown in relation to the largest class. For example, in the first stage, when Dense, Spread and Differentiated are combined against Debris, the difference between the total number of images in each class is made up by adding generated Debris images to the real Debris training dataset. **At every stage, the smaller, single class is balanced against the larger aggregate**

class using generated images, creating proportional image classes, in order to counteract the problem of data imbalances.

The temporal constraints imposed in this method focus on the overall morphological class relationships within the dataset. This contrasts with using dynamic cellular changes between video frames directly in model training. **Instead, the temporal relationships between image classes are exploited to improve the efficiency of network learning for classification of individual images based on cellular morphology.** The details of this method are described below.

Firstly, Debris cell colonies are separated from the other three classes, which are grouped together into a single class and sent to the next stage of classification. In the second stage, differentiated cells are separated from a grouped class of Dense and Spread cells. Finally, in the third stage Dense and Spread cells are separated into their individual classes. Performing classification in this manner allows for the exploitation of the natural relationship between classes, as well as allows for more fine control of dataset augmentation using generated images based on class imbalance and dataset proportions.

In the final stage of classification, the power of generated features for dataset augmentation is explored by testing the saturation point for the Dense vs. Spread classes. Thousands of generated images are added to both the Dense and Spread classes and a plot of network performance vs. augmentation level is created for visual reference. Empirically, it is shown here that this combination of hierarchical classification and dataset balancing using generated image augmentation outperforms a four-class CNN configuration using various standard dataset balancing methods, and that there is a positive correlation between the

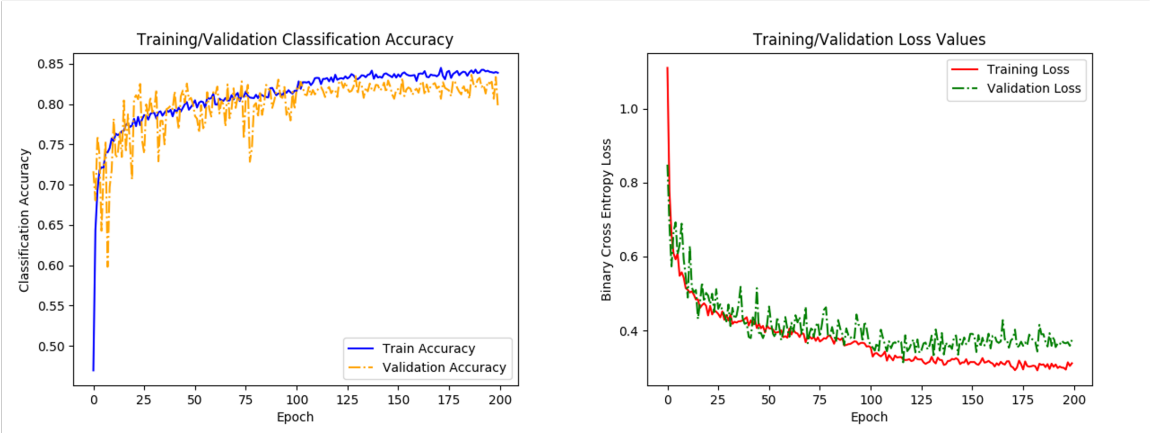

Figure 3.5: Graphs of network accuracy (left) and cross-entropy loss (right) for training and validation datasets. A small respective bump/dip in accuracy/loss is observed at 100 epochs, where the learning rate parameter is reduced. Training levels out before the 200 epochs, indicating that the network has finished learning.

level of generated image augmentation and the classification accuracy of the network, up to a saturation point.

All models are trained for 200 epochs (training slows down at this mark), using Cross-entropy loss criterion, and Stochastic Gradient Descent optimizer (LR: 0.005, momentum: 0.8, weight-decay: 0.0001, batch-size: 64), where the learning rate is reduced by half, half-way through training. All configurations are trained with 5-fold cross validation, using an 80:20, train:test split. Figure 3.5 displays the classification accuracy and loss parameters observed over the course of training and is used to check for network overfitting and training efficiency. The following section provides specific details about the dataset used in this work, as well as the results of CNN training using augmented datasets from GAN.

3.5 Results and Discussion

3.5.1 Data

Data for this study come from experiments performed by Dr. Barbara Davis in the laboratory of Dr. Prue Talbot. They are aimed at determining the effects of nicotine exposure on diseased, induced pluripotent stem cells (iPSC) expressing the Huntington's Disease (HD) Phenotype. HD is a progressive neurodegenerative disorder that affects motor neurons in the adult brain [57, 58]. Nicotine has been shown to have a neuroprotective effect on neurodegenerative diseases such as Parkinson's Disease [59]. The theory of this work is that nicotine may have a similar affect in HD nueronal growth and development.

To test this hypothesis, HD iPSC's are exposed to nicotine at varying concentrations (Control, 10^{-4}M, 10^{-5}M) in-vitro over the course of a 48 hour culture period. Large size, (2908×2908), images of cellular culture dishes are collected at 10x magnification using the Nikon Biostation CT, an an automated incubator-microscope unit. Collected images contain thousands of colony areas at various stages of the developmental process and require pre-processing and manual annotation to be used to train the deep learning networks. These images are pre-processed in accordance with the approach described in Section 3.4.1. The resulting image dataset is then manually annotated for ground-truth as described in the following section.

3.5.2 Ground-truth Validation

A breakdown of number of images has been shown in Table 4.1. Images were sorted based on pre-determined, visually distinct features presented in each class that cor-

respond to phenotypic differences. Morphological classes were determined by the experts who collected the data to best reflect the phenotypic characteristics of distinct cell colonies. Ground-truth data was obtained via manual annotation by an expert researcher (A.W.) for the entire dataset. These annotations were validated by two additional researchers (G.P., R.T) on a random subset of image data to provide consensus and check for variability and subjectivity. The original sorting was confirmed by training a neural network on each of the three annotated data subsets and determining which sorting provided the best testing results on a class-wise basis. The rationale for the patch-based sampling approach used for this work is presented in the next section.

3.5.3 Patch-based Sampling

There are two main reasons why a patch-based sampling approach is used in this work. The first reason is that image classes in this dataset are based on morphological phenotype. These morphologies are determined by their sub-colony level texture patterns, and image patches provide a local view of these morphologies. Also, the contiguous nature of colony morphology means that multiple classes can appear in a single colony image, the use of image patches provides the most reliable means of limiting class overlap when performing patch-based classification.

The second reason is that GAN training is a data intensive process, with the counterproductive goal of increasing dataset size. Therefore, a patch-based sampling method is implemented for the following reasons:

Class	# Samples
Debris	3587
Dense	3934
Diff	656
Spread	10506
Total	18683

Table 3.5: Data Breakdown for Four Morphological Classes. Class imbalances observed here are a factor of the natural growth and differentiation cycle of the cells.

- to increase the apparent training dataset size
- to accommodate efficient network architectures (it is widely recognized that GANs are effective when images are relatively small ($\leq 64 \times 64$) but are prone to mode collapse with high-resolution images)
- to standardize input size, as image crops vary in dimension
- to model low-level features (i.e., fine-grained textures), which show high variation across image patches for a given class
- to increase general variability via patch sampling, which generally improves training
- to aid in the analytical goal of classifying contiguous, multi-label cell colonies in a patch-wise manner using only cellular morphology

To this end, the GAN model is used to generate 64×64 patches of colony images. Image quality is measured using various standardized qualitative and quantitative measurements as described in the following section.

3.5.4 Assessment of Generated Image Quality

There are several methods by which generated image quality is measured in this work, including visual appearance, quantitative scores, and efficacy in application for the desired task. Figure 3.6 displays real and generated image patches across the four morphological classes in the dataset for the various GAN configurations.

Visual assessment of these samples reveals that the generator has been able to capture both the general structure and fine-grained morphological features of the cell colonies, and that these features are distinguishable for the individual cellular phenotypes. Some configurations, such as dcGAN, show greater image variation and visual quality than others, such as auxGAN, especially for the Differentiated class. Images generated with the multi-class generator used in auxGAN shows signs of feature entanglement, where image features from one class are present in another. This is because the generator has difficulty separating these features using a single model. Other methods, such as wGAN, fail to generate any realistic images, such as for the Differentiated class, where only noisy black images can be seen.

While the generator may be able to convince the trained human eye of its ability to produce realistic images, that does not necessarily mean that it will be able to provide useful information to the learning task. For this, feature based quality measures are required to determine the level of relative image realness with respect to the real dataset.

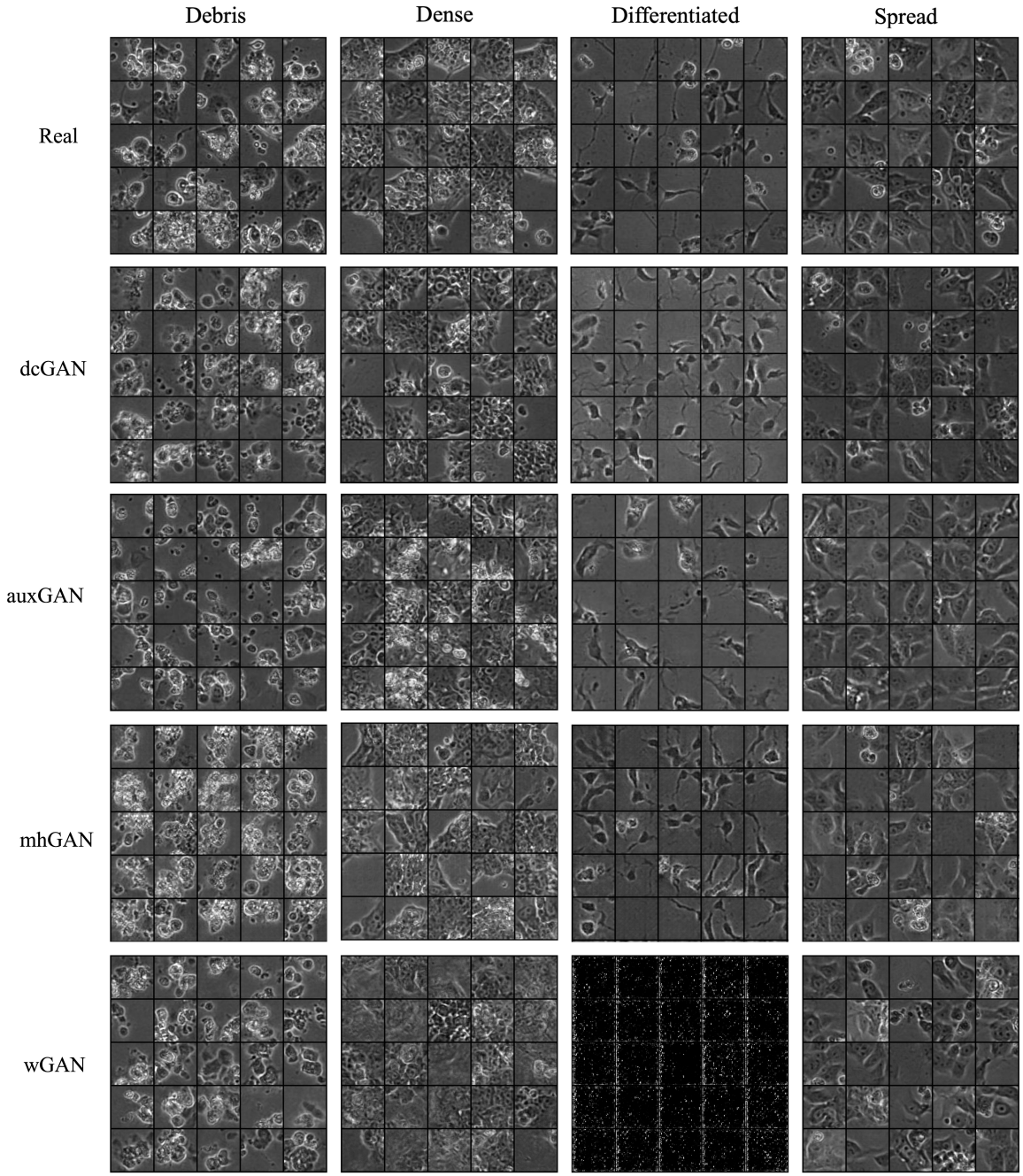

Figure 3.6: Image patch samples for real and generated images. A comparison of class-wise image features displays generally realistic image features indicative of morphological class. However, visual appearance of images provides only a qualitative measure of image quality, where quantitative metrics are necessary to determine image realness.

Inception Score

Inception score is a quantitative measurement of generated image quality with respect to image classification based probability distributions [60, 55]. This method combines measurements of generated image realness and variability based on the output predictions of a pre-trained Inception network. The output is an entropy based score using the Kullback-Leibler divergence (D_{KL}, Equation (3.12)) between the class-wise generated image distribution, P, and the overall generated distribution Q, where x is the discrete probability and X is the probability space. In this work, Inception score is used to monitor network training, and determine the iteration at which GAN training is optimized. The network representative of this inception score is chosen as the generative model for dataset augmentation. The inception score has a range based on the number of classes in the dataset, for example, from 0 to 1 for binary classification:

$$D_{\mathrm{KL}}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{3.12}$$

Frécet Inception Distance

The Frécet Inception (FID) is another image quality metric that seeks to overcome some of the shortfalls of the Inception Score by directly comparing generated image distributions to corresponding real image distributions [56]. FID score is calculated using the mean and variance of the output feature maps of the Inception network given real and generated image inputs. This method provides a calculation of the image quality with respect to the real images, and is sensitive to noise, blur, and occlusions.

In this work, FID score is used to evaluate the "realness" of generated image datasets on a class-wise basis. Together, the Inception score and FID create a comprehensive view of generated image quality and can be used to determine the efficacy of different generator configurations. Observations of image quality made using these metrics are provided in the following section.

3.5.5 GAN Training Visualization

During GAN training, the Inception and FID scores for generated image patches is tracked and generated images are gathered at various time points for visual reference, Figure 3.7. Many aspects of network learning can be inferred from the observation of these visual references over time. At the start of training (Epoch 1), the Generator begins to distinguish between foreground and background, displaying only gray level splotches of indistinguishable colony areas. Images begin to show more features resembling realistic colony patches around the 50-epoch mark, displaying more distinct, lower contrast splotches that also include the high-intensity bright areas indicative of the 'halos' observed in the experimental dataset. These high-level morphological details are seen across the entire dataset and indicate that the network learns high-level features first.

The progression of displayed features indicates that training is stable, but the network has not yet captured the fine-grained texture and gray-level variation that is indicative of the morphological class. These features become more distinct by Epoch 100 as cell boundaries can be seen, and contrast intensifies. Inception score reaches a precipice

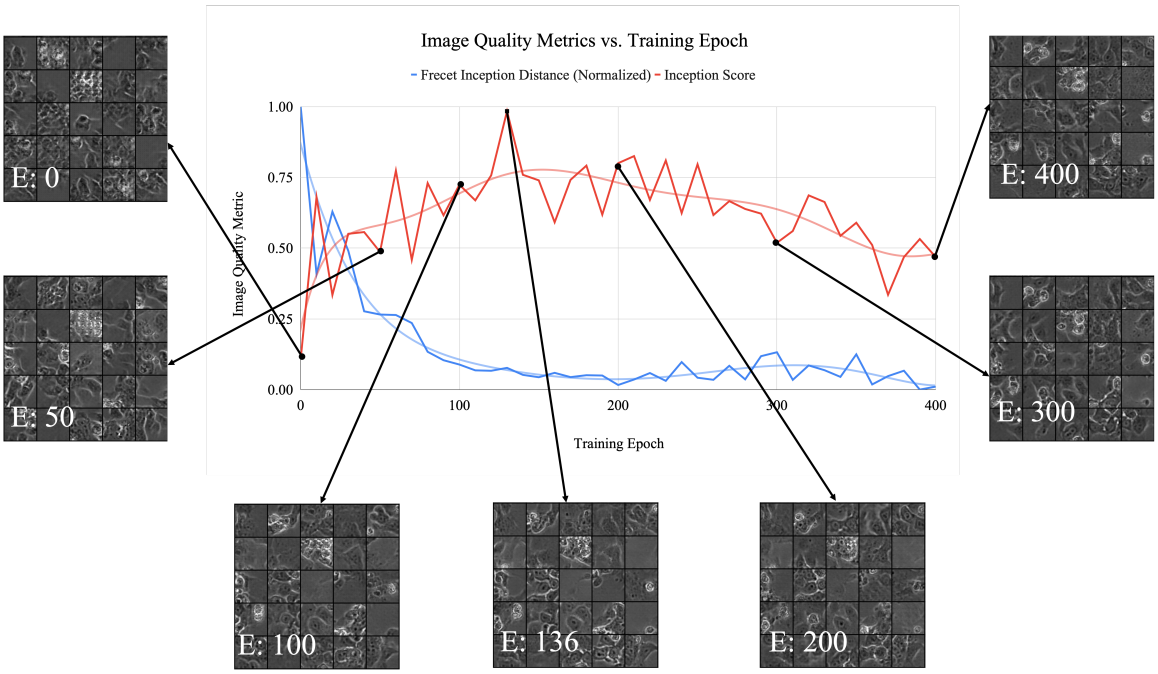

Figure 3.7: Normalized Generator inception score (red) and FID (blue) per training epoch with example images at various intervals for the Spread Class. Graphs include accompanying trend-line. Training epoch numbers are marked by a white 'E' in the bottom of each image. Agreement between Inception score and FID can be seen in terms of their relative minimum and maximum values versus training epoch.

Image Class	Optimal Generator Epoch	Inception Score
Debris	116	2.60
Dense	444	2.32
Diff	225	2.38
Spread	136	2.57

Table 3.6: Epoch values and corresponding inception scores at which the GAN generator is determined to be optimally trained, based on the plot of inception score vs. training epoch. Each GAN is trained on an individual class, and, therefore, requires a different level of training based on the number of images in each class, as well as complexity of features and other variables. The optimal GAN is used to generate images for each class to be used for dataset augmentation.

at the 136-epoch mark (similarly the FID score reaches a valley) and although the GAN continues to produce realistic looking images, it can be noted that the Inception score and FID do not improve as training continues past that point.

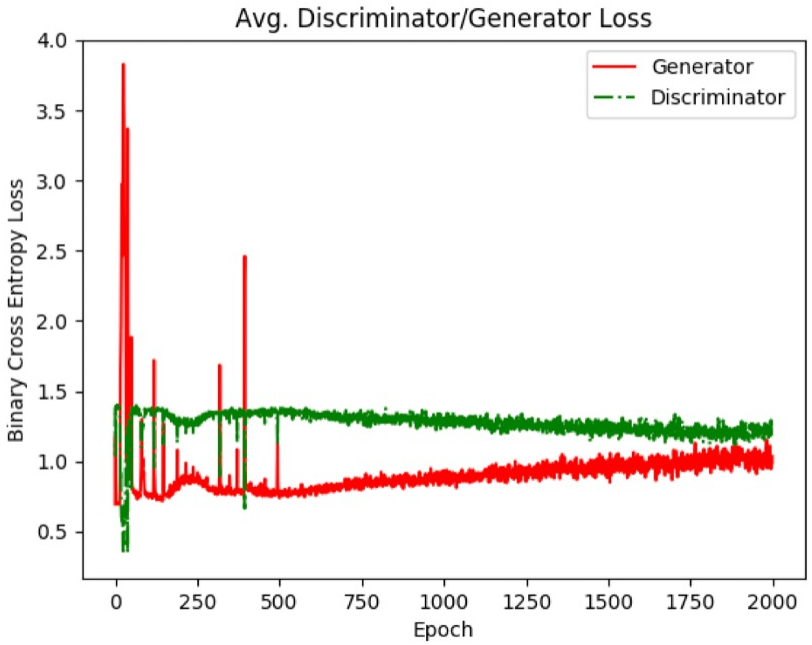

Figure 3.8: Generator and Discriminator Loss values for the Dense class. As training progresses, the GAN reaches an equilibrium which is when training is considered finished. Using individual GAN models for each image class allows the GAN to be trained for different amounts of time based on the image class.

This plot is useful as a deterministic way of choosing the most realistic generator. Realistic colony features such as clear boundaries between cells, and bright white 'halos' rounded and single cell objects can be seen in the final generated images. Training is considered finished after the GAN loss values have reached equilibrium (Figure 3.8), however, the training epoch at which the most realistic images are generated is determined using the inception score. The number of epochs required to reach equilibrium is subject to the size of the dataset on a class-wise bases, as well as other variations between image data classes. The epoch values at which the inception-score for each class is optimal is shown in Table 3.6, along with their corresponding Inception Scores at that point. Once the generator is trained for each class, images are generated by simply providing random noise vectors to

each of the generator and saving image outputs. In the following section, the ability of different GAN networks to produce realistic images is explored using the defined quality metrics.

3.5.6 GAN Network Comparisons

In this work, multiple GAN configurations are compared to determine the most effective method of image generation, in terms of image realness and variability. The FID (Frécet Inception Distance) score is used as a quantitative measure of these traits, in a class-wise manner, as shown in Table 3.7, where a lower FID score is better. The different GAN configurations have various potential uses in terms of generative feature learning. For instance, the wGAN configuration is known for preventing mode-collapse, whereas the mhGAN is designed to select the most relevant images from the input dataset in terms of the discriminator's decision output, and for this reason it may be better at modeling multi-modal distributions.

Figure 3.9 displays these values graphically, and shows that the dcGAN+MSE configuration produces the best (lowest) FID score for all classes. The reason for this is that using the entropy loss as a regularizer during GAN training provides additional information to the network about the real image distribution. Subsequently, this allows for the generation of more relevant images in terms of image appearance and variation, as measured by the FID score.

Other configurations, such as wGAN, are not appropriate for this dataset because the trained dcGAN model is not subject to mode collapse, as evident by the variation present

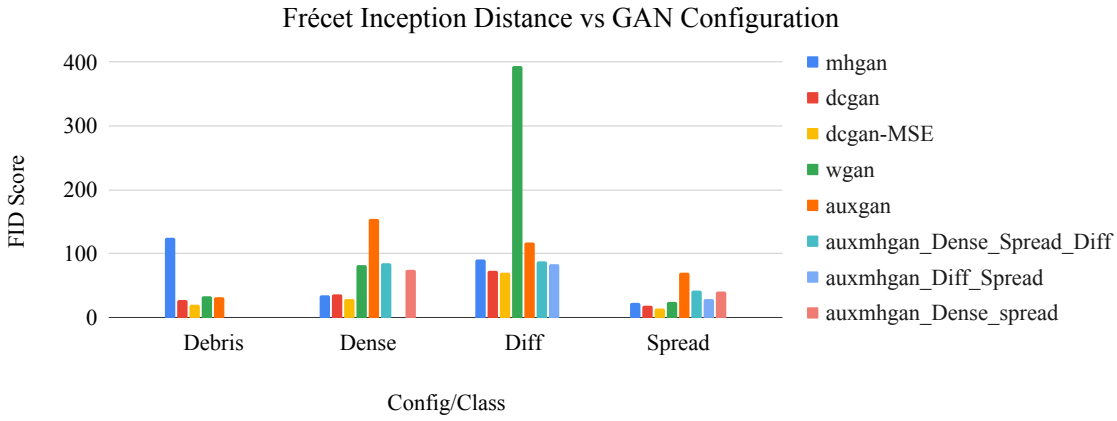

Figure 3.9: Bar Graph of Network Configuration vs FID Score by Image Class. The dcGAN+MSE configuration consistently displays the best performance in terms of this metric.

Config./Class/FID	Debris	Dense	Diff.	Spread	Average
dcGAN	27.73	36.32	72.77	18.51	38.83
dcGAN + MSE	19.5	29.5	70.7	13.67	33.34
wGAN	33.85	81.45	393.94	24.05	133.32
auxGAN	31.62	155.13	117.92	69.86	93.63
mhGAN	125.22	35.03	90.63	23.05	68.48
aux-mhGAN (Dense, Diff, Spread)	x	84.93	88.7	41.53	71.72
aux-mhGAN (Dense, Spread)	x	x	83.37	29.55	57.54
aux-mhGAN (Diff, Spread)	x	74.0	x	41.05	56.46

Table 3.7: Frécet Inception Distance (FID) Scores for each GAN Configuration by Image Class. An x indicates where the GAN was not trained to generate the specific image class.

in the generated images. For the differentiated class, wGAN is unable to generate relevant images, and produces only image artifacts, hence the relatively high FID score for this class. The reason for this is that wGAN reduces the apparent image variation of this already small class by restricting the feature weights. The mhGAN configurations do not produce higher quality images, in terms of FID score, and takes far longer to train than a standard dcGAN (on the scale of days). mhGAN reduces both the overall number of image samples available to the generator, resulting in lower quality images with less variation. Auxiliary conditional-

GAN applications (auxGAN, aux-mhGAN) are not as effective at producing the individual image classes using a single model as are individual GAN's trained on a single class. This is because they are susceptible to feature-entanglement, which is the inevitable sharing of input features based on the latent representation of the input data distribution [61].

The ultimate test of the efficacy of the proposed method is to perform generated image augmentation on CNN networks. For this, the dcGAN+MSE generator configuration is used to generate images in line with the temporal classification method as described in Section 3.4.4. The following section discusses the results of the classification networks trained using the generated dataset augmentation scheme.

3.5.7 Classification Metrics

Classification results for this study are measured using the True Positive Rate (TPR, Equation (5.3)) and F1-Scores (Equation (3.14)). TRP, also known as recall, is a measure of the sensitivity of the classifier, where TP is the number of correctly classified positive instances, and FN is the number of incorrectly classified negative instances. F1 is the harmonic mean of precision, which is the TP over the sum of TP and False Positives (FP), to recall, and considers the false positives and false negatives equally in its calculation. These metrics provide information about the classifier's ability to accurately separate positive and negative instances and are used to compare CNN training configurations, as presented in the following section:

$$TPR = \frac{TP}{TP + FN} \tag{3.13}$$

Config./Class/TPR (Std.)	Debris	Dense	Diff.	Spread	Average
Unbalanced	0.9141 (0.0144)	0.8093 (0.0211)	0.8807 (0.0342)	0.9144 (0.0093)*	0.8789
Sampler Balanced	0.8570 (0.0184)	0.9300 (0.0189)	0.9274 (0.0219)	0.8410 (0.0073)	0.8888
Weight Balanced	0.9030 (0.0312)	0.8065 (0.0249)	0.8439 (0.0715)	0.9300 (0.0290)	0.8708
Generator Balanced	0.9105 (0.0206)	0.7940 (0.0116)	0.8999 (0.0247)	0.9172 (0.0124)	0.8804
Temporally Balanced	0.9277 (0.0148)	0.8157 (0.0142)	0.8856 (0.0289)	0.9646 (0.0040)*	0.8984

Table 3.8: Class-wise True Positive Rate for four-class CNN with and without dataset balancing. Several variations of balancing are used here, the most effective of which is supplementation using generated images in line with the temporal training configuration proposed in this paper. The p-value indicated by the $*$ is calculated using the student t-test and is equal to 3.9×10^{-6}.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.14}$$

3.5.8 Dataset Balancing Using Generated Image Augmentation

To test the efficacy of generated image augmentation as a dataset balancing technique, several standard methods of balancing are compared to the generated augmentation method. These techniques include the use of a weighted cross-entropy loss function and an image sampling technique called Imbalanced Dataset Sampler from Ming Yang (repository has over 1400 stars on GitHub) [62].

Table 3.8 demonstrates that the Generator balanced dataset configuration outperforms both traditional sampling techniques. This is because sampling and weight balancing inherently detract from the learning ability of one class in favor of another, whereas supplementing the dataset with generated images not only leaves the higher data class

intact but adds useful features to the smaller data classes in the process. Furthermore, these classification scores are improved when balancing data classes in coordination with the temporal classification scheme proposed in this paper.

Exploiting the temporal relationships between image classes increases the average true positive rate of classification is by approximately 2% in the "Temporally Balanced" training configuration over the "Unbalanced" configuration. This increase corresponds to approximately 200 images out of a 10,000-image dataset. In a high-throughput scenario, where millions of images can be collected over the course of a single experiment, this 2% increase can have a substantial effect on the outcome of experimental findings. These hierarchical results are detailed below.

3.5.9 Effect of the Temporal Classification Scheme

The efficacy of using the temporal relationships between image classes to inform network training is explored via a hierarchical CNN classification scheme. Tables 3.9-3.11 detail the classification metrics in relation to the three stages of temporal classification. First, images are separated into viable (Dense/Spread/Diff.) vs. unviable (Debris) classes, which serves to remove colony areas containing dead cells. Areas of Debris can be observed within larger colonies of viable cells, however, removing these areas during the first stage of classification negates the possibility of these misclassifications happening in the downstream stages.

Config./Class/TPR (std.)	Debris	Dense/Diff./Spread	Average
Unbalanced	0.9145 (0.0097)	0.9570 (0.0058)	0.9357
Generator Balanced	0.9277 (0.0148)	0.9545 (0.0053)	0.9411

Table 3.9: Hierarchical Tier 1: True Positive Rate for temporal combination of Viable Cell Classes vs. Debris Cells. This stage acts as a filtration step to remove unviable and unhealthy colony areas.

Config./Class/TPR (std.)	Diff.	Dense/Spread	Average
Unbalanced	0.8792 (0.0255)	0.9941 (0.0007)	0.9367
Generator Balanced	0.8856 (0.0289)	0.9935 (0.0007)	0.9396

Table 3.10: Hierarchical Tier 2: True Positive Rate for separation of Dense/Spread classes from Differentiated. This tier serves to remove the mature cell colonies from the early and intermediate stage classes. The Dense/Spread classes have the highest level of misclassification, due to their relative proximity in terms of the downstream differentiation process, and subsequent similarity in texture features.

Config./Class/TPR (std.)	Dense	Spread	Average
Unbalanced	0.8187 (0.0140)	0.9624 (0.0035)	0.8906
Generator Balanced	0.8157 (0.0142)	0.9646 (0.0040)	0.8902

Table 3.11: Hierarchical Tier 3: True Positive Rate for classification of Dense vs. Spread. The balancing of the Dense Class, using generated images, in relation to the Spread class shows slight improvement over the unbalanced configuration, and marked improvement over the four-class configuration.

Config./Class/F1 (std.)	Debris	Dense	Diff.	Spread	Average
Unbalanced	0.8732 (0.0059)	0.8430 (0.0082)*	0.8580 (0.0164)	0.9119 (0.0036)**	0.8715
Generator Balanced	0.8599 (0.0099)	0.8599 (0.0050)*	0.8714 (0.0009)	0.9433 (0.0030)**	0.8836

*: p-value = 4.32×10^{-3}; **: p-value = 3.88×10^{-7}

Table 3.12: F1 Score for Unbalanced and Generator balanced temporal training configurations show an overall improvement for the balanced configuration on average, as well as statistically significant increases in the Dense and Spread classes. The p-value indicated by the $*$ is calculated using the student t-test.

It can be seen in Table 3.9 that the balanced configuration for this first stage improves the true positive rate for the Debris class by 1.32% over the unbalanced temporal configuration, and 1.36% over the unbalanced four-class configuration (Table 3.8).

The second stage of classification sends the three viable classes to be separated into Differentiated vs. Dense/Spread. This stage is useful in distinguishing between the late-stage adult cells represented by the Diff. class, and the early/intermediate stages (Dense/Spread respectively). The Dense and Spread classes are the most closely related in terms of developmental stage, and therefore have the most similar morphological features, and are misclassified most often given the four-class classification scheme.

The second stage of temporal classification, Table 3.10, shows an increase in the true positive rate of classification for the Differentiated class over the unbalanced temporal configuration by 0.64%. This value represents a smaller increase in comparison to the other classes, which may be due to the relatively low amount of image data for the differentiated class that is available for training GAN and for classification. When looking at the entropy histograms in Figure 3.3 and overlap values in Table 3.4, it can be seen that the differentiated class has the lowest overall overlap, which may contribute to the lower improvement in performance.

Additionally, this stage filters out 99% of the Dense and Spread images, which lends merit to the hypothesis that these classes are most closely related. This stage seeks to use the temporal relationships between early and middle stage colonies and the differentiated colonies as a marker for the classification boundary of these images. Finally, the Dense and Spread images are sent to the final stage of classification, where they are separated into their individual classes.

The final stage of temporal classification, Table 3.11, displays an increase in classification rate for the Spread (∼5%) over the unbalanced, four-class configuration, as shown in Table 3.8. This improvement is shown to be statistically significant (p-value = 3.9×10^{-6}) using a student t-test. This improvement is also noteworthy given the difficulty of separating these two very similar classes.

Table 5.4 displays the F1-score classification results for the unbalanced configuration and the generator balanced temporal configuration. This metric shows improvements for the Dense, Differentiated, and Spread classes in the balanced configuration, with an average improvement of approximately 1%, and statistically significant increases in the Dense ($\sim 2\%$) and Spread ($\sim 3\%$) classes, using the student t-test. These results further illustrate the predictive power of generator augmented, temporally constrained CNN configurations in relation to straight forward classification networks.

Up to this stage of classification, no generated images have been added to the Spread class. As an additional test of the efficacy of generated features, the saturation point of generated image augmentation, (i.e., the point at which the accuracy no longer increases with increasing number of generated images) is empirically determined using the Dense and Spread image classes.

3.5.10 Saturation Point of Generated Image Augmentation

In addition to the problem of imbalanced datasets, dataset limitations represent another problem in the field biological image classification. These limitations are due to the difficult nature of biological experimentation, in terms of time, money, and experimental yield. Often image datasets are relatively small in terms of the amount of data needed

Figure 3.10: Graphs of classification metric (Left: True Postitive Rate, Right: Classification Accuracy) vs. number of added generated images for the Dense and Spread classes. These graphs are used to determine the saturation point of a CNN, which is where the generated images no long provide useful features to the model.

to efficiently train a neural network. Therefore, generated image augmentation could be a viable means of increasing the apparent dataset size and available features for deep learning applications.

To test this theory, generated images are added to the Dense and Spread classes to train a two-class CNN until the classification accuracy no longer improves with increasing number of generated images. This point represents the saturation point of the network, where the generated images are no longer providing more useful features to the model. This test also serves as a method of determining the effectiveness and practicality of dataset augmentation using generated image features because it directly compares the number of added images to the improvement in performance.

Figure 3.10 details the results of these experiments. The graphs of generated image augmentation show a positive linear relationship for both the Dense and Spread classes (in terms of TPR and Classification Acc.) until about the 10,000 image mark, where improvement falls. This could be indicative of the network overfitting on generated

images and failing to learn the features of real images. While these improvements are small in comparison to those of the dataset balancing experiments, they confirm that limited dataset supplementation is a viable avenue for generated image augmentation.

3.6 Conclusions

The temporally constrained, generative dataset augmentation scheme employed in this paper represents an improvement in the performance of deep learning algorithms for classification tasks involving limited and imbalanced biological image datasets. Known GAN methods are compared to generate images, and then image-class relationships are employed to design a temporal classifier. The method shows classification improvements for all image classes, as measured by true positive rate and F1-score, without sacrificing the performance of any class. Moreover, this work highlights the importance of exploiting domain knowledge in similar tasks, which in this case comes in the form of the temporal relationships between image classes. While deep learning has become the gold-standard in terms of image feature representation and learning, it is not a substitute for prior biological knowledge. To this end, one item of future work will involve the expansion of this work to include video data to incorporate cellular dynamics. Overall, the combination of domain knowledge with deep learning is paramount for the effective modeling of biological image features, and it allows for the synergistic design of deep learning-based algorithms, and experimental data collection.

Chapter 4

Triplet-net Classification of Contiguous Stem Cell Microscopy Images

4.1 Abstract

Cellular microscopy imaging is a common form of data acquisition for biological experimentation. Observation of gray-level morphological features allows for the inference of useful biological information such as cellular health and growth status. Cellular colonies can contain multiple cell types, making colony level classification very difficult. Additionally, cell types growing in a hierarchical, downstream fashion, can often look visually similar, although biologically distinct. In this paper, it is determined empirically that traditional deep Convolutional Neural Networks (CNN) and classical object recognition techniques are

not sufficient to distinguish between these subtle visual differences, resulting in misclassifications. Instead, Triplet-net CNN learning is employed in a hierarchical classification scheme to improve the ability of the model to discern distinct, fine-grain features of two commonly confused morphological image-patch classes, namely Dense and Spread colonies. The Triplet-net method improves classification accuracy over a four-class deep neural network by $\sim 3\%$, a value that was determined to be statistically significant, as well as existing state-of-the-art image patch classification approaches and standard template matching. These findings allow for the accurate classification of multi-class cell colonies with contiguous boundaries, and increased reliability and efficiency of automated, high-throughput experimental quantification using non-invasive microscopy.

4.2 Introduction

Stem cells are a widely used, recapitulative, in-vitro (outside of the organism) model for human growth and embryonic development [7]. These cells have the ability to reproduce biological processes such as differentiation, or the downstream lineage change of cells and cell colonies in response to intrinsic and environmental factors. Differentiation occurs naturally throughout the human life cycle but can be affected by environmental factors causing genetic mutations which result in developmental diseases [10], [63], [64]. Experimental efforts to determine the mechanisms by which these changes occur often involve the observation of cellular behavior using non-invasive phase-contrast microscopy. This method of data collection avoids the need to sacrifice cells for staining or mechanical sorting, allowing cellular changes to be observed over time. Dynamic cellular and colony

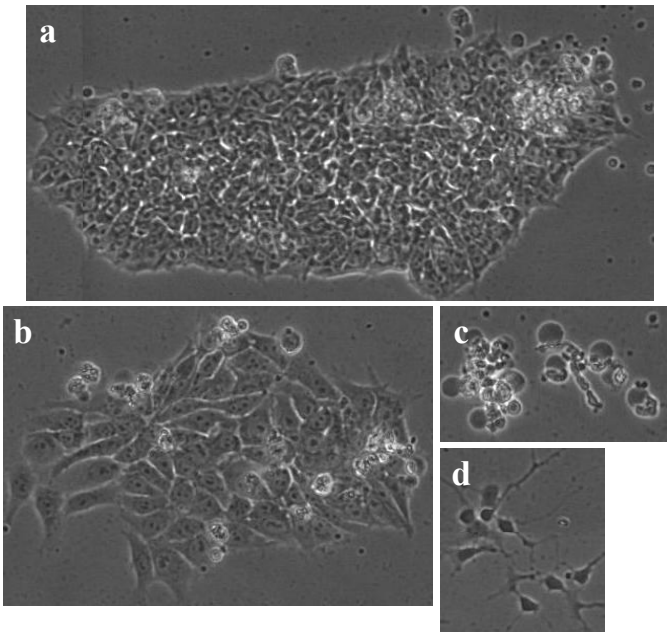

Figure 4.1: Example image crops for the four classes used in this work. (a) Dense cells are compact colonies made up of small, pluripotent stem cells. (b) spread cells are less compact than dense cells, and are made up of progenitor cells, a downstream intermediate of differentiated cells. (c) The Debris class represents cells that are unhealthy or dead/dying and assume a rounded morphology. (d) Differentiated cells are neuron like formations that have dark cell bodies and thin, spine like axons. These are mature cells that are the final downstream endpoint of differentiation.

morphology can be a direct indication of cell health and developmental status, and for this reason, can be used to quantify cellular experiments [15].

Image analysis is commonly supported by video-bioinformatics programs aimed at leveraging cell colony features for classification and quantification of experiments [16]. These programs use hand-crafted or learned features in combination with machine learning classifiers to distinguish between different cell types in an image. This task becomes increasingly difficult with high volumes of data containing multiple types of cells within a single image. Previous work has attempted to address this issue by training a classifier to

distinguish between each cell type in an image separately, however changes in morphology can be extremely subtle, and are difficult even for sophisticated deep learning models to classify. These subtleties are related to the hierarchical nature of differentiation, during which precursor cells turn into adult cells, often through intermediate stages known as progenitors [65]. Differentiation can occur at various rates, proportions, and locations within a cell colony, making it difficult to quantify visually.

In the brain, differentiation of adult neural stem cells occurs on a continuous basis in both a generative and regenerative manner [66][9][67]. Neurogenesis occurs mainly in the Hippocampus, which is the part of the brain associated with memory and spatial awareness, at a rate of approximately 700 neurons per day in adults [68][69]. The differentiation process can be altered by diseased states and other external factors affecting normal cellular development. For example, the progression of the Huntington's disease (HD) model used in this work can alter both the appearance and behavior of neurons in the human brain [70]. Another manifestation of HD includes Hippocampal decline, and associated symptoms of these changes in patients[71]. In-vitro for HD use visual biomarkers and developmental timelines as an indication of disease progression. Previous studies using HD stem cells have outlined the normal, in-vitro differentiation process and have shown that the maturation of differentiated neurons is delayed under the diseased state [72].

One potential avenue for Huntington's disease therapeutics is with nicotine agonists. Nicotine and nicotinic-acetylcholine receptor agonists have an effect on the symptoms of Huntington's disease in both experimental and clinical settings [73][74][75]. Furthermore, nicotine also affects the function of the Hippocampus in a variety of ways depending on

dosage [76][77]. The relationship between Hippocampal function, adult neurogenesis, Huntington's disease progression, and nicotine exposure is a potential avenue for understanding HD mechanisms and possible therapeutics.

In-vitro stem cell models provide an avenue of high-throughput experimentation. Behavioral observations captured via light microscopy and experimental evaluation using automated computer vision and deep learning algorithms are crucial for analytical standardization. These programs efficiently model relationships between the morphological appearance of image classes and their developmental status and can help delineate disease mechanisms when designed in coordination with experimental endpoints. Exploitation of domain knowledge during algorithm development and implementation can improve model learning and efficacy and help to streamline the analytical pipeline.

For example, mathematical models of neuronal stem cell differentiation have elucidated mechanisms of down stream lineage commitment that occur as a result of non-Markovian stochastic processes [13]. These in-vitro studies highlighted that the transitional phase from the pluripotent to intermediate state occurs rapidly than from intermediate to the neuronal stage. This paper exploits inter-class relationships observed during stem cell differentiation, and uses them to classify image patches using Triplet-net. Network learning is guided by biological constraints inspired by these downstream lineage changes.

The approach outlined in this paper combines a Convolutional Neural Network (CNN) and a Triplet CNN network in a hierarchical manner to classify four distinct morphologies from contiguous, HD expressing, induced pluripotent stem cell colonies (iPSC) undergoing differentiation during exposure to nicotine. Image patches containing morpho-

logical texture patterns, representative of four distinct classes, are extracted from segmented colony images observed via non-invasive phase-contrast microscopy. Patches are first classified using a three-class CNN that combines two of the most closely related classes, Dense and Spread, after which a downstream Triplet-net CNN is used to sort these images into their respective sub-classes.

The main advantages of the Triplet-net method include the use of feature comparison for fine-grained feature learning, as well as the avoidance of data-imbalances by using Triplet-pairs for network training. The functionality of the triplet-net method in this work is to reduce feature entanglement between the Dense and Spread classes that are the most closely related biologically and morphologically. The discriminative power of the triplet-net allows for more accurate classification of image patches with fine-grained texture features.

One caveat of the Triplet-net method is determining the most effective use of learned features for image classification during network testing. While triplet-net learning is effective for feature representation, classification can take many forms. Multiple validation methods are compared in this work including non-linear classifiers and template matching. The most straightforward method of validation is a non-linear multi-layer perceptron (MLP) for feature classification, however it is shown empirically here that template feature matching using pre-selected anchor images provides the most accurate results as opposed to training an additional classifier on learned features.

By-hand selection is the gold standard for choosing anchor images for the template matching method. For this, anchor templates are removed from the dataset before training by a skilled worker to determine representative images from the dataset. This process

is not repeatable for new anchor selection or generalizable for other datasets, especially those involving biological images. There is a need to circumvent or otherwise automate this process. Therefore, a standardized, morphological and texture feature-based method is employed in this paper to automatically determine representative anchor images from the dataset. This anchor selection method is compared to a CNN trained with a joint Triplet-softmax loss function that allows straightforward image classification. The use of anchor images for triplet-net testing improves classification by directly comparing image feature embeddings to make final class predictions. Furthermore, the proposed method is compared to two alternative state-of-the-art approaches to demonstrate the power of triplet-net learning to distinguishing between visually similar classes.

4.3 Related Work

Many works have utilized CNN's in biological image classification tasks for their state-of-the-art performance and feature learning abilities. For example, Waisman et al. [78] use a Resnet50 (He, et al. [79]) CNN architecture to determine differentiation status of mouse embryonic stem cells using only morphological features from light-microscopy images.

In addition, Buggenthin et al. [80] use CNNs to predict the lineage commitment of differentiating hematopoietic (blood) stem cells based on their morphological behavior. They use a large dataset of single cell images to train a combination of a CNN and recurrent neural network (RNN) to output a lineage score based on morphology and use temporal information to improve prediction robustness. Similarly, Campanella et al. [81], use multiple instance learning (MIL) to perform clinical pathology predictions for whole-slide cancer

histology images. This method combines ranked image patch selection and RNN feature aggregation to perform prediction for various cancer types.

In a previous work, multi-label classification of differentiating HD iPSC's was performed using CNN in a non-invasive manner. Cellular sub-classes were detected using a patch based, sliding window classification method for images containing more than one class. From these predictions, colony area per class was plotted over time to determine growth and differentiation rates in response to nicotine exposure. Several conclusions were drawn based on these observations, including that experimental growth rates were higher than those of the control, and that downstream classes presented themselves on the borders of cell colonies, which is consistent with biological knowledge [82]. However, it was noted in this work that the patch size for these image patches (224×224) was too large to accurately separate between classes within colonies, such that there was some overlap between classes within the image patch [5].

In all of these works relatively straightforward implementations of CNN/RNN were used to classify microscopy images based on a learned multi-modal distribution of input data. However, these approach are sometimes ineffective to determine features of very closely related classes. To solve this problem, Triplet-net configurations can be used as a method to improve learning by using representative images to reduce feature entanglement and inform model decision making. These networks have commonly been employed for tasks such as person re-identification in security footage [83], when trying to determine if an image is one of two similar looking subjects. However, there exist some implementations of Triplet-net for biological image analysis.

Although Triplet-net CNNs have been used much less for biological image classification, they show promise as a method for distinguishing between visually similar classes, performing semi-supervised learning, and for overcoming data imbalance problems commonly associated with biological datasets. For example, Schubert, et al. [84] use a Triplet-net to learn representations of 3D electron microscope renderings of neurons. They use softmax trained on learned feature embeddings to classify glial neurons and used this information to aid in image reconstruction. This work used Triplet-nets to learn 2d representations of 3d input volumes, and perform secondary classification using the feature embeddings.

Gupta et al. [85] leverage Triplet-loss to determine mitotic events in human epithelium cells observed via fluorescence microscopy. They train a Support Vector Machine (SVM) classifier on feature embeddings learned by a CNN using Triplet-loss and outperform a CNN trained using cross-entropy classification loss in terms of the Matthews correlation coefficient. Their work uses Triplet-networks to learn representative features of whole, single-cell images obtained via fluorescence microscopy. While this work addresses the problems of data-imbalances and learning fine features, it implements SVM for classification, which is a generally less powerful classification technique, and requires additional training on top of feature selection. In the proposed work, Triplet-net is used to directly classify a diverse dataset of gray level images using a patch-based method. Additionally, stem cell colony images require unique considerations for feature learning and classification, such as contiguous, and multi-label images.

Other examples of triplet-net for biomedical images include Chen et al. [86] who generate retinal artery masks using U-Net style GAN with and accessory triplet-loss module to improve mask generation. This work leverages the triplet loss function to improve the performance of a GAN, but does not directly apply triplet learning to image classification. Lei et al. [87] classify skin disease images using triplet-CNN with a novel class-center loss function to help combat class imbalances within a small dataset. This class-center loss is used to constrain the learned feature distribution of an image dataset with high variability, whereas the focus of the proposed method is to distinguish between images with very similar feature distributions. Thammasorn et al. [88] use triplet-CNN feature extractor to perform classification of gamma radiation images. They compare the discriminative power of different machine learning classifiers trained on learned features against standard radiomic features.

Further implementations of triplet-net include, Sarhan et al. [89] who perform detection and segmentation of retinal microaneurysms using multi-scale, patch-based, U-Net with a triplet-network refinement module. This method uses triplet-loss as a secondary comparison module to improve segmentation, as opposed to directly classifying the images. Huang et al. [90] propose a novel batch-based triplet-loss function to improve feature extraction for classification of medical image datasets using 'light-weight' CNN. This implementation leverages triplet-net features to train a soft-max classifier for direct classification aimed at combating the over-fitting problem associated with CNN's trained on small datasets. They combine the learned feature embedding with the encoded ground-truth label to build a similarity matrix to compute the batch-loss. They determine the effectiveness of

their method on chest x-ray and rash image datasets and show improvements over several baseline 'light-weight' CNN implementations. They use their novel loss function during the training process, but resort to soft-max classification during testing, whereas the method in this work directly employs the triplet-net comparison for image classification.

Another work that leverages triplet-loss is by Theagarajan et al., [45] who employs Triplet-net to train a CNN for classification of stem cell images. Their CNN classifies whole cell images containing an entire single cell or cell cluster with similar local visual features. In contrast, the proposed work seeks to distinguish between patches of stem cells within a large contiguous stem cell colony with very subtle differences in fine-grained visual appearance, which adds a higher level of difficulty to the classification. To the best of our knowledge, this is the first approach that uses deep learning to classify patches of stem cells within a contiguous stem cell colony. Additionally, the proposed approach differs from [45] because the classification decision is made by measuring the distance between the learned distributions of anchor and query input images, instead of using a non-linear classifier to perform two-class classification.

One major limitation of the proposed triplet-net classification method is the process of anchor image selection. This is especially important for biological datasets, because it requires a skilled worker to choose representative images by-hand. This process is biased and error prone even when a skilled worker is available. Anchor images have been used in the past for tasks such as biometric authentication, and automated anchor selection methods have been suggested to overcome this pitfall. For example, Uludag, et al. [91] test various methods of template selection including distance measurement and clustering.

Thomas et al., [92] provides another method of image classification that directly uses template matching for biological image classification. They employ multi-template matching via normalized correlation coefficient for the detection of zebra fish embryos in microscopy images. However, these anchor selection methods have not been employed to tasks involving deep learning.

All of the aforementioned methods circumvent the use of anchor images for template feature comparison by performing auxiliary classification using learned feature embeddings. In this paper, the proposed method is compared to various triplet-net configurations similar to those mentioned above as well as two alternative approaches, namely [92] and [81]. Furthermore, the feature matching based image classification scheme using automated anchor image selection algorithm described in this work shows improvement over other triplet-net derived feature classifiers and is potentially generalizable to other datasets while removing sources of human error.

4.3.1 Contributions of this Paper

In light of the state-of-the-art, the contributions of this paper are as follows:

- The first deep learning approach that uses a biologically inspired, hierarchical Triplet-net classifier to identify stem cells across four distinct morphological phenotypes within large contiguous colonies

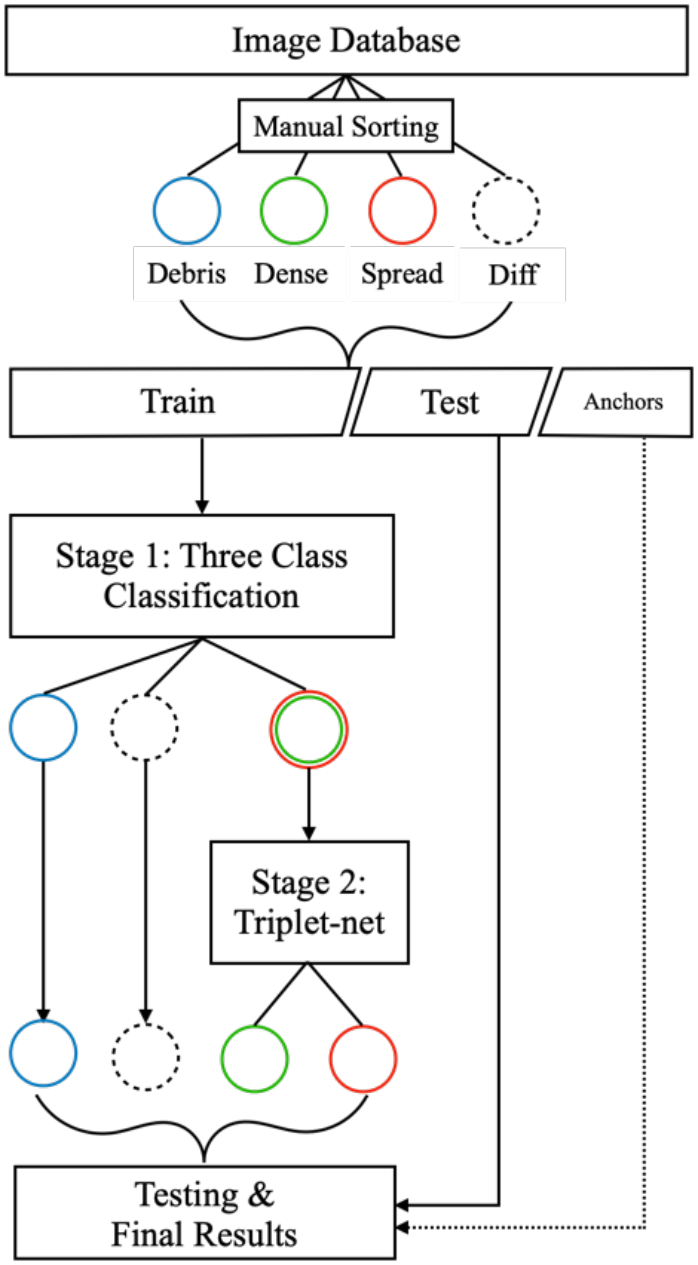

Figure 4.2: Overall Diagram for Training and Testing. Before training, images are sorted by hand into four classes. A portion of this dataset is used to train a three-class CNN that filters out two visually similar classes into a single class which is filtered once more into individual classes by the Triplet-net, which leverages anchor images to improve classification performance. During testing, the "Test" portion of the dataset is used to determine the accuracy of the approach as compared to the ground-truth and anchor images are used to classify images from the second stage of prediction using the Triplet-net method.

- Trains Triplet CNN to learn fine-grained features of stem cell colonies to distinguish between images that have very subtle visual differences. Addresses the challenges of image dataset imbalances with the goal of performing non-invasive experimental quantification
- Overcomes the limitation of Triplet-net anchor image selection using an automated, texture feature-based clustering algorithm that outperforms hand-selected anchor images
- Empirically shows, via comparison, that traditional CNNs do not adequately learn the subtle visual differences between patches of stem cells within a large contiguous stem cell colony

4.4 Technical Approach

Figure 4.2 describes the hierarchical classification approach presented in this work. In the first stage, image patches are sorted into three classes, namely: Debris, Dense/Spread, and Differentiated. Dense and Spread images are grouped together because they are visually similar to each other (see Figure 4.2). Images that are classified as Dense/Spread are then fed into a second stage Triplet-network, that uses a Triplet-loss function to learn fine-grained features that can distinguish between these two closely related classes. The reasoning behind this hierarchical system is that the Dense and Spread stem cell images are closely related biologically, and therefore share similar morphology [72]. The goal of the hierarchical approach is to improve the overall classification accuracy in comparison to a straightforward four-class CNN.

4.4.1 Image Pre-processing and Ground-Truth

Image patches are used in this work for the task of multi-class, morphological-based classification for the following reasons:

1. they contain texture patterns that are representative of the individual stem cell classes on a local level

2. detected colony Regions-of-Interest (ROI), (see Figure 4.1) are of various sizes and aspect ratios

3. using a standard size input (128×128) circumvents having to resize images. This image patch size is specifically determined based on the microscope objective magnification and camera resolution within the Nikon Biostation CT unit. The combination of these hardware specifications results in the size of an image pixel being 0.8 μm/pixel, and making the height of a 128 pixel patch equal to 102.4 μm. This patch size allows for image patches to fully capture texture patterns of the various cell colonies which contain individual cells that range between 10-100 μm [93]

4. increasing the variability of input images helps to prevent over-fitting. These colony images are first detected via entropy-based segmentation [5], from large (2908×2908) raw, phase contrast microscope images containing multiple colony ROI's, and then manually sorted into four morphological classes representing four distinct phenotypes.

Class	# Samples
Debris	3587
Dense	3934
Diff	656
Spread	10506
Total	18683

Table 4.1: Data distribution of the images for all of the four-classes in the dataset

Morphological Class	Visual Description	Phenotypic Implications
Debris	"bubble" like cells or cell aggregates, and small, dark circular objects surrounded by bright white rings denoting a rounded appearance	These cells are unhealthy, dying cells which round up and float to the tops of colonies, or can form whole colonies of dead cells
Dense	compact colonies with rounder, relatively small cells, no visible nuclei, no clear distinction between cells; more uniform appearance and texture	These cells are indicative of human pluripotent stem cells, which tend to grow in tight colonies and are smaller than mature cells
Spread	larger, more spread out cells and cell colonies, flatter appearance with visible nuclei and more distinct cell boundaries; less uniform texture, some more elongated cells displaying protrusions	Spread cells are considered progenitor cells that represent an intermediate stage between the Dense and Differentiated cells. They often occur on the outer edges of Dense colonies and relate closely to both of its adjacent classes
Differentiated	dark black cell body area with thin, spiny, protruding axons. Individual cells or groups of cells connected by axons	Differentiated cells are mature neurons, the downstream endpoint of the differentiation process

Table 4.2: Morphological Class Descriptions

This ground-truth processing was performed by a skilled biologist trained in cell culture, and was guided by a list of visual features for each of the classes as described in Table 4.2.

Examples of colony images for each class are shown in Figure 4.1. During every training epoch, a random image patch, of size 128 × 128, is selected from every input image and used to train a CNN to perform classification; examples of which can be seen in Figure 4.3. During testing, center crops of the same size, taken around the middle pixel of every input image, are used as samples to evaluate the accuracy of the trained CNN. Classes are balanced using class weights based on their proportion in the training dataset. More details about the training and testing procedure employed in this work are provided in the next section.

4.4.2 CNN Training

A standard, 19-layer VGG style network is used as the architecture for all CNN configurations in this work [94]. VGG was empirically pre-determined to provide the best results for this classification task over Resnet50 and other standard architectures. This network uses all 3×3 convolutional kernels with one pixel each for stride and zero-padding. Each layer consists of one convolutional module, batch normalization, and a rectified linear activation unit (ReLU). Max pooling layers are used in down sampling layers, and the largest number of feature maps in any layer is 512. The classifier consists of two fully connected linear layers, both with ReLU, and implement dropout at a rate of 50% to reduce over-fitting.

The CNN classification network is trained for 200 epochs using the stochastic gradient descent algorithm (SGD, Learning Rate = 0.005, Weight Decay = 0.0001, Momentum = 0.9), where all network training hyperparameters are determined empirically using net-

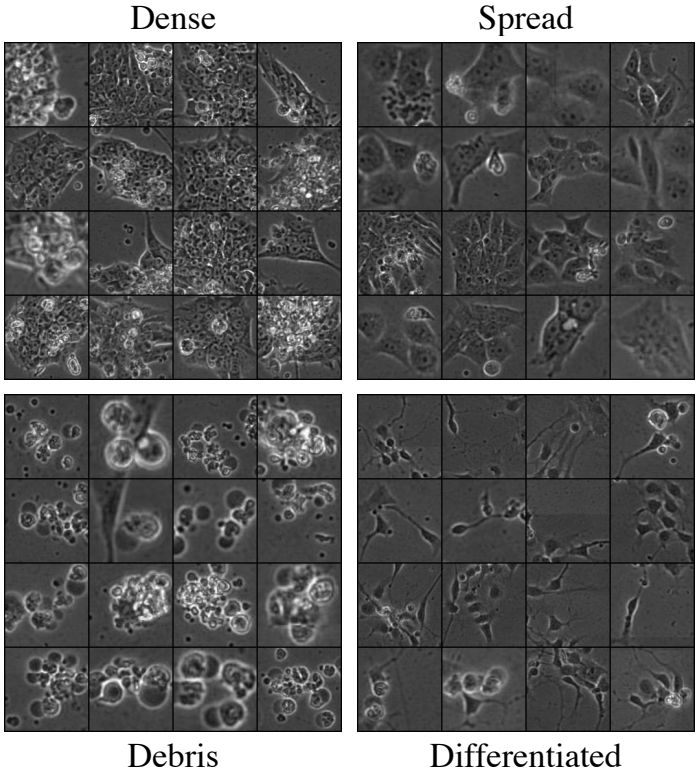

Figure 4.3: Example image patches for training the CNN. During every epoch, random image patches of size 128×128 are selected from each training image, and used to train the network, as described in Section III (B). The local texture features present in image patches are representative of the global features used to determine the classes for this work. Visual similarities between patches provide the reasoning behind implementing a Triplet-net approach for the most closely related classes.

work optimization search, and the learning rate is reduced by a factor of 10 after 100 epochs. The network is optimized with a weighted cross-entropy criterion as shown in Eq. (4.1), which takes the decision level loss as input to inform the SGD algorithm. In Eq. (4.1) C is the total number of classes, α_i is the class weight, y_i is the label of ith the class, and p_i is the predicted probability of the ith class. All results are averaged over the networks trained using 5-fold cross-validation, with an 80:20, train:test, data split for each cross-fold. Class weights are inversely proportional to the number of images in each class.

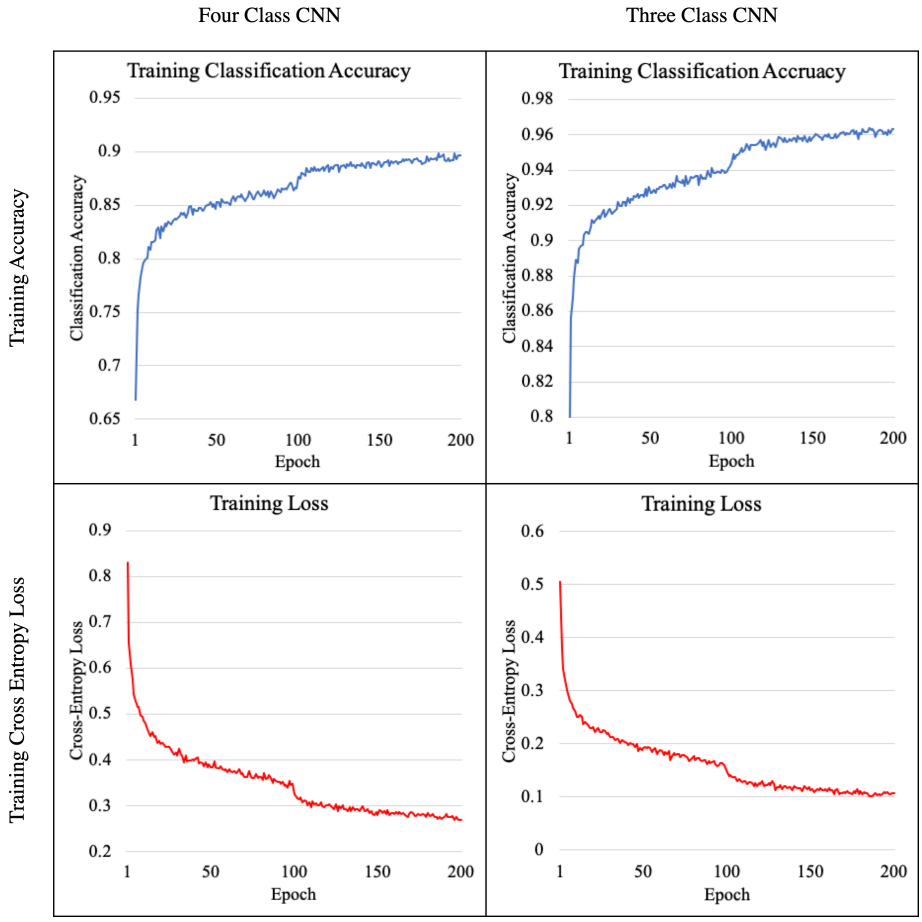

Figure 4.4: Comparison of CNN training between three and four-class CNN configurations. As expected, the three-class CNN trains more efficiently than the four-class CNN as indicated by the increase in training accuracy and decrease in training loss in relation to the curves for the four-class configuration.

$$L_{Entropy} = -\sum_{i}^{C} \alpha_i \, y_i \, log(p_i) \tag{4.1}$$

Both a four-class and a three-class CNN configuration are trained using this network, where the three-class configuration combines the Dense and Spread classes into a single class, which is then sent to the downstream Triplet-net. Comparison between the

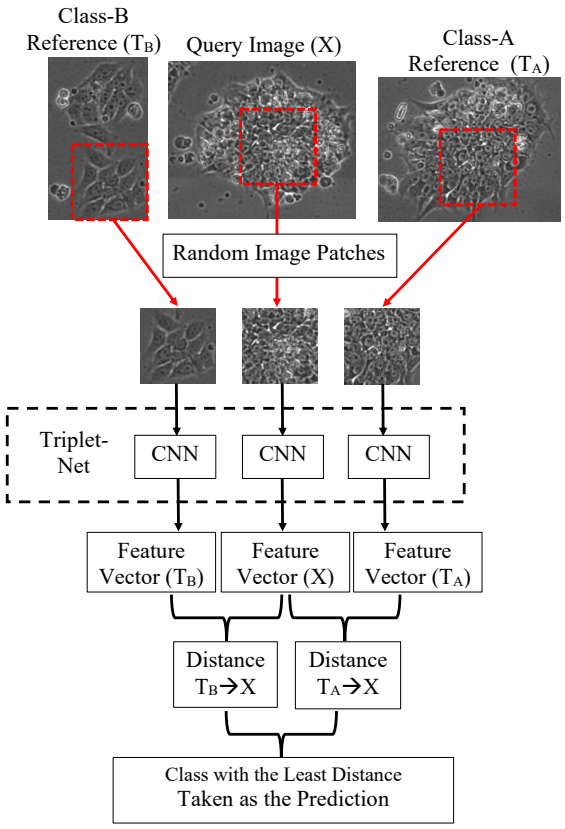

Figure 4.5: Triplet-net schematic used for the Dense and Spread classes. The input to the Triplet CNN for training is a Triplet-pair made up of a query image, one positive and one negative reference image (T_A, T_B). The Triplet-net then makes a classification decision based on the distances between the feature maps of the query image and each of the anchor images. In this way, the network is provided with more information during classification, versus straightforward classification based on the learned image distribution. During testing, the anchor images are used to compare the feature distance of the query image and a majority vote is taken across all anchor images.

training accuracy and loss curves between the four and three-class CNN configurations, shown in Figure 4.4, confirm that the three-class CNN trains more efficiently than the four-class CNN, in terms of both training accuracy and cross-entropy loss; approximately 0.90 versus 0.96, and 0.25 versus 0.10, respectively. This is due to the fact that the model has to learn a less complicated distribution for three classes as compared to four. This also

lends merit to the hypothesis that the Dense and Spread classes contain similar features that are able to be modeled by the CNN as one class. All CNN's are tested with the same size center crops (128×128) of testing images for consistency. After the images are sorted into three classes by the first stage CNN, images that are classified as Dense/Spread are sent to a Triplet-net for a second stage of fine-grained classification.

4.4.3 Triplet-net Training

Triplet CNN is a sub-class of CNNs known for extracting fine-grained features while simultaneously maximizing the interclass variance and minimizing the intraclass variance [95]. During training, the Triplet CNN takes as input a query image and one reference image from each class (one positive and one negative reference). The output of the Triplet CNN consists of the two pairwise Euclidean distances between the extracted features for the query image and the two reference images as shown in Figure 4.5. For a correct classification, the pairwise distance between the query image and the reference image belonging to the same class must be smaller compared to the distance between the query image and the reference image belonging to the opposite class.

All of the Triplet CNN configurations in this approach are trained for 15 epochs using 100,000 randomly selected Triplet pairs for every epoch. The Triplet CNNs were optimized using the SGD algorithm with the Ranked Marginal loss function as shown in Eq. (4.2). In Eq. (4.2), T_A and T_B are the two anchor images and $G(T_x)$ is the pairwise distance between the feature extracted by the Triplet CNN for the query image and the anchor image, where x is either A or B. In Eq. (4.2) if $Y = 1$ it indicates that the anchor, T_A, belongs to the same class as the query image, whereas, $Y =$ -1 indicates that the anchor

image, T_B, belongs to the same class as the query image. The value of the margin is set as 1 for all experiments. Similar to training the CNN, all results are averaged over the networks trained using 5-fold cross-validation, using a 75:25, train:test data split.

$$L_{Triplet} = Max(0, -Y * (G(T_{\mathrm{A}}) - G(T_{\mathrm{B}})) + margin) \tag{4.2}$$

4.4.4 Anchor Image Selection for Triplet-net

During testing, the average pairwise Euclidean distance is computed between a given query image, X, and each of the 11 pre-selected anchor images from each class, and the class with the least average pairwise Euclidean distance is taken as the final classification. 11 images are selected in line with [96], where 10 anchors are used. In this work, anchor images are selected and removed from the entire dataset before performing the training:testing split using the remaining images. This way the anchor images are never seen by the network during training, and can be compared directly to the testing images during evaluation in an unbiased manner. However, this number of anchor images is not necessarily effective across all datasets, so the effect of changing the number of anchors images on the variance of classification accuracy is tested in this paper in Section 4.5.1. The minimum number of anchor images for improved performance is determined experimentally by varying the number of anchor images used during the testing phase. It should be noted that selecting the anchor images is very important because they can affect the final classification results if chosen incorrectly.

Normally, anchor images are hand-selected based on the requirements fulfilled by each images appearance in accordance with the class descriptions in Table 4.2 in the

supplemental material [97]. Exemplary images from each class, displaying the most uniform, clearest texture patterns, based on these criteria, are removed from the dataset and used as the anchor images for all of the experiments. Hand-selected anchor images for each class are shown in Figures 4.6 and 4.7.

However, this selection process is subjective and not repeatable or generalizable, especially for complex cellular image datasets. Therefore, an automated morphological and texture feature-based machine learning method is employed here to select image anchors based on feature clustering. To select the image anchors for each class, first, morphological (cell colony aspect ratio (1) and solidity (1), [4]) and texture (local binary patterns (10) [98], gray level co-occurrence matrix features including contrast (8), dissimilarity (8), homogeneity (8), and angular second momentum (8) (ASM) [99] , and Segmented Fractal Texture Analysis (SFTA)(48) [100]) features are calculated and concatenated to create a 92-dim feature vector for each image (the number of features are denoted after each feature name in parentheses). Vectors were normalized between 0-1 for each individual feature as well as the entire concatenated vector. These feature vectors were then used to select exemplary images by using one of the following methods, where the goal of each method is to select N=11 anchor images from the dataset of Triplet-net images containing all samples of the Dense and Spread classes:

1. *Random Selection*: N samples are selected at random for each class across the entire Triplet-net dataset. This method serves as a baseline for comparison for the effect of anchor selection on testing images.

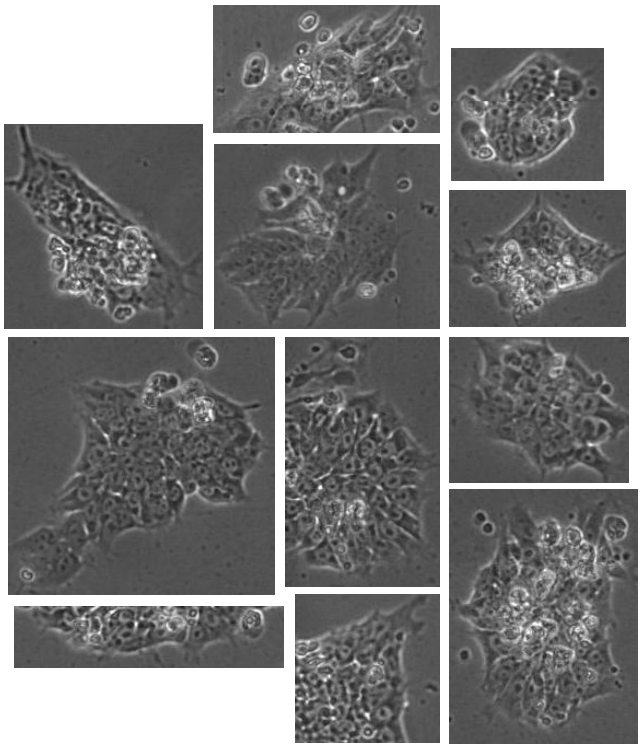

Figure 4.6: Hand-picked Dense Anchor Images

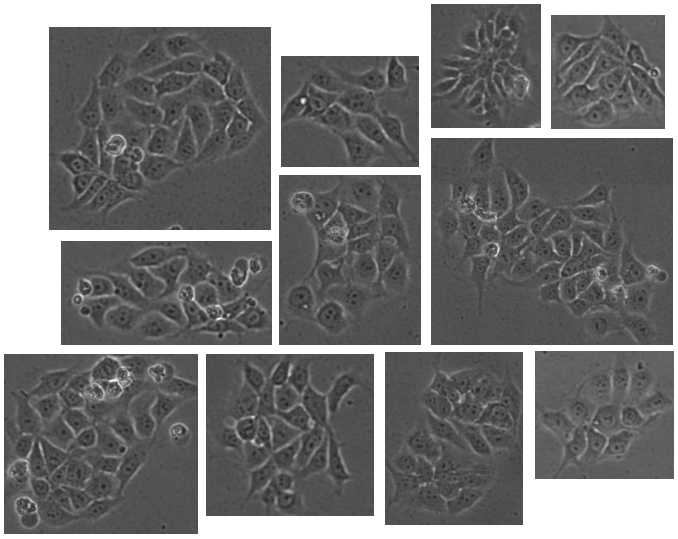

Figure 4.7: Hand-picked Spread Anchor Images

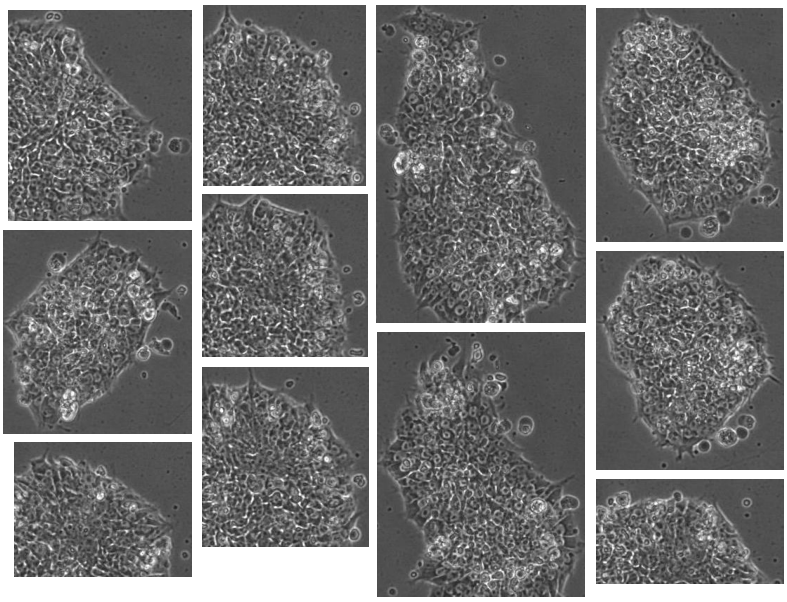

Figure 4.8: Automatically selected Dense Anchor Images

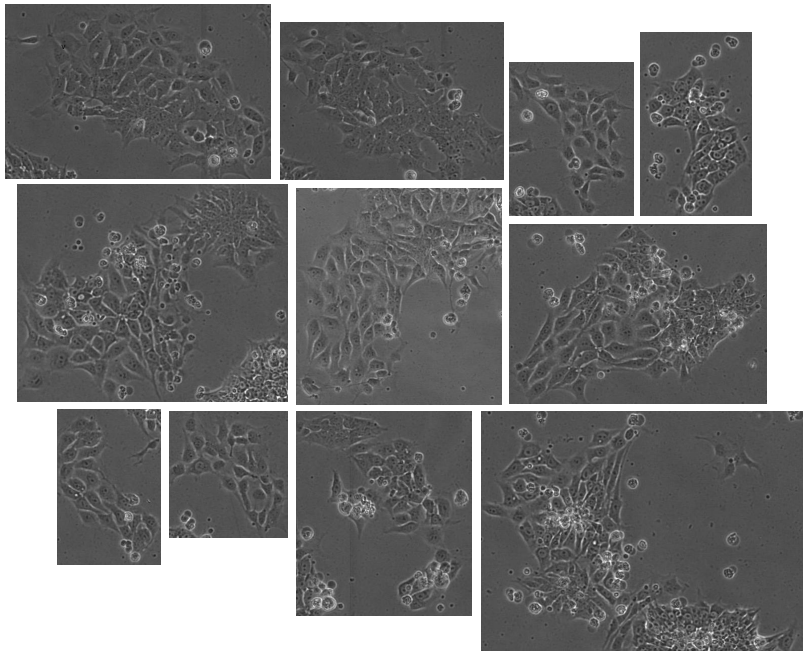

Figure 4.9: Automatically selected Spread Anchor Images

2. *Exhaustive Max-Correlation*: The Pearson correlation coefficient is used to exhaustively compare every image in the dataset for each of the two relevant classes. The average correlation of each image with every other image of the same class is used to build a ranked list of images with the highest correlation to the rest of the dataset. The top N samples are selected from the ranked list and taken as anchor images. This method represents the anchor images that show the closest similarity relationship to every other image in the dataset.

3. *K-means (K=2)*: K-means clustering algorithm is trained on feature vectors for the dataset across the two Triplet-net classes (Dense and Spread). The N closest images to each cluster center (in terms of Euclidean distance), given that the image that belongs to the cluster center representing a class, are taken as anchor images. This method assumes that clusters representing each class contain feature vectors that are representative of the entire dataset.

4. *K-means (K=11)*: K-means clustering algorithm is trained on feature vectors for the dataset for each Triplet-net class separately. The feature vectors are used to train two K-means clustering algorithms, one for each class, and iterated into K clusters. K=11 for this method such that the single feature vector that is closest to each of the K clusters is taken as one of the N anchor images. This method assumes that enough variation exists within the dataset such that each cluster is representative of a subset of images that contains certain distinct features.

5. *Explicit Selection Via Linear Discriminant Analysis (LDA)*: LDA is a supervised algorithm used to maximize the scattering between image classes (S_B) while minimizing

the scattering within (S_W) each class across all features representing the dataset of Triplet-net images. The goal of LDA is to maximize Fishers discriminant, which is shown below in Equation 4.3 [101][102]:

$$F(W) = \underset{W}{argmax} \frac{W^T S_B W}{W^T S_W W} \tag{4.3}$$

where the Fisher's discriminant F is a projection matrix that maximizes the ratio of the determinant of the scattering matrices $\frac{|S_B|}{|S_W|}$ using the transformation matrix W. From the training classification results, a ranked list of probability scores for every image per class is used to determine the top N images for each class to be used as anchors. This method is used as an explicit means for finding the closest feature vectors to each class mean, that are also the furthest from the opposite class and assumes that the feature space is linearly separable.

Example images for by-hand and automatically selected anchors are visible in Figures 4.6, 4.7, 4.8 and 4.9. As a comparison between the automatic anchor selection methods and the hand-selected anchors, multiple Triplet-network configurations are tested using anchors selected by each of the methods. These testing configurations are used to determine the image features that are most representative of the image dataset, and to determine the effects of anchor image selection on model generalizability. These feature selection methods can potentially be applied to other datasets where a skilled worker is not available to select anchor images. These results are presented in Section 4.5.1.

4.4.5 Triplet-softmax Training Configuration

To determine the effect and necessity of anchor images on classification performance, the standard Triplet-net training/testing configuration was also compared to a compound, Triplet-net/softmax configuration. In this network, an extra fully connected, softmax layer is added to the Triplet-CNN and the loss function is modified to include the cross-entropy loss criterion from the two-class softmax classification. The new aggregate loss, $L_{agg} = L_{triplet} + L_{softmax}$ is used to update the network during training, and during testing, the softmax classification layer can be used to directly determine input image classes without the use of anchor image comparison. These results can be found in Section 4.5.1.

4.4.6 CNN Classification Results

Data

Data for these experiments were collected by the laboratory of Dr. Prue Talbot, in the Department of Cell, Molecular, and Developmental Biology, at the University of California, Riverside, CA. The goal of the experimental work was to determine the effects of nicotine on the progression of a neurodegenerative disorder called Huntington's Disease (HD) [57] [58]. The mechanisms by which this disease causes its Parkinson's like symptoms are still not well understood, but it has been shown that nicotine can help with Parkinson's symptoms by acting on Nicotinic Acetylcholine receptors in the brain [59].

To test this hypothesis in vitro, Huntington's disease expressing induced pluripotent stem cell (iPSC) colonies are cultured under exposure to nicotine, and their behavior

over time is compared to those cells that are under non-toxic conditions to determine if nicotine has any neurogenic or neuroprotective effects. These behavioral responses are observed via time-lapse, phase contrast microscopy imaging, and a dataset of 15, 48-image sequences is collected at 10x magnification. As stated above, colony ROIs are segmented from the large size (2908×2908) raw images, and 128×128 patches are extracted from these colony images for training and testing of neural networks. Example image patches are shown in Figure 4.3 and a breakdown of the number of image examples per class is shown in Table 4.1.

Four-Class CNN Results

Multi-class CNNs are the standard in image classification for biological datasets and are now commonly used in many applications including medical and biological image classification. As a baseline for classification accuracy, a 4-class VGG19 CNN is trained as described above, and the results are evaluated using the true positive rate (TPR), true negative rate (TNR), and F1 classification scores (Table 4.4.6). This network achieves a true positive rate of 90.44% and 90.92%, respectively for the Dense and Spread classes. A representative confusion matrix for this network is shown in Table 4.4, which highlights the two most commonly confused classes, Dense and Spread. The Spread class is confused as Dense 118 times and the Dense class is confused as Spread 66 times.

As a test of the effect of network architecture on classification accuracy, the Resnet50 CNN is trained and tested in an identical manner and results are compared to

Class/Metric (std.)	Debris	Dense	Spread	Differentiated	Average
VGG19 - TPR	0.8793 (0.0039)	0.9044 (0.0259)	0.9092 (0.0144)	0.9425 (0.0218)	0.9088
VGG19 - TNR	0.9769 (0.0025)	0.9466 (0.0119)	0.9198 (0.0173)	0.9948 (0.0009)	0.9595
VGG19 - F1	0.8938 (0.0057)	0.8623 (0.0063)	0.9228 (0.0033)	0.9089 (0.0126)	0.8969
Resnet50 - TPR	0.8967 (0.0153)	0.8084 (0.0251)	0.9356 (0.0179)	0.8662 (0.0113)	0.8767
Resnet50 - TNR	0.9694 (0.0051)	0.9708 (0.0070)	0.8719 (0.0170)	0.9971 (0.0004)	0.9523
Resnet50 - F1	0.8894 (0.0045)	0.8503 (0.0077)	0.9154 (0.0039)	0.8962 (0.0095)	0.8878

Table 4.3: Average of 5-fold Classification metrics for the four-class CNN

	Predicted Class			
True Class	Debris	Dense	Diff.	Spread
Debris	642	19	4	65
Dense	15	707	0	66
Diff.	0	0	119	9
Spread	54	118	14	1903

Table 4.4: Confusion Matrix for Four-class CNN

Parameter	Value
Minibatch Size	36 images
Storage	8.55 Gb
No. Params.	31.84 M
MACs	7.33 G
Train Time/Epoch	759.63 s
Test Time/Epoch	170.94 s

Table 4.5: Network Complexity for Triplet-net CNN

the VGG19 configuration in Table 4.4.6. Table 4.5 displays the computational complexity associated with the triplet-net training on a single Nvidia GeForce RTX 3080 GPU. Despite the value of their relative depth, these networks, in their current configurations, contain a similar number of total parameters (31.8 M v. 25.9 M, respectively). It is observed that on

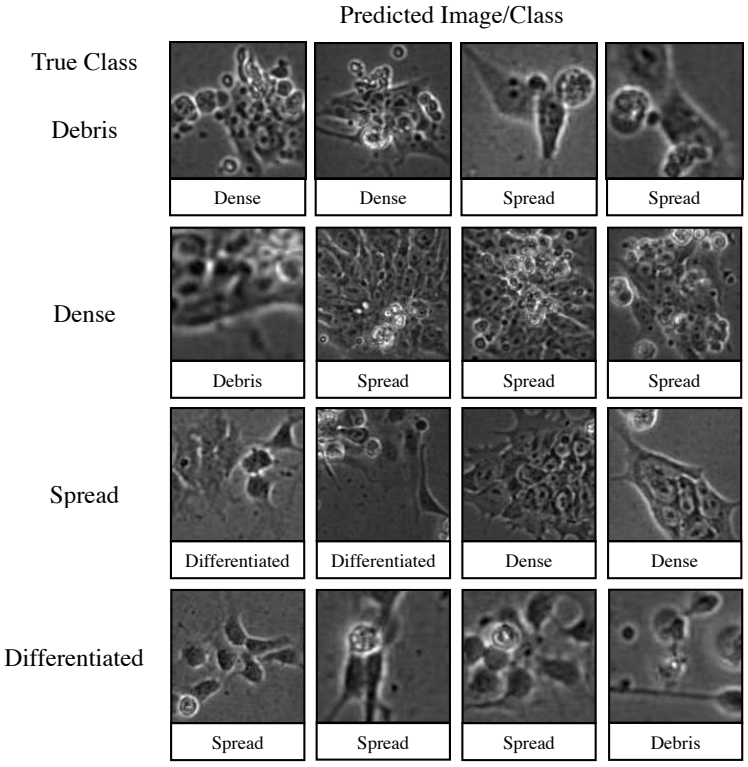

Figure 4.10: Misclassified image patch samples for the four-class CNN configuration. These misclassifications highlight the need to implement a Triplet-net for distinguising between visually similar classes. The most commonly confused classes are the Dense and Spread classes, because both colonies grow in relatively tight morphological conformations, and represent closely related phenotypes, biologically.

average VGG19 outperforms Resnet50 in every metric (TPR: 3.120%, TNR: 0.7200%, F1: 0.9100%), as well as for the majority of individual classes. This discrepancy in classification accuracy may be due to the influence of 'receptive field' of the network with respect to feature mapping, and is a factor of the network depth, kernel, and convolutional parameters. [103]. The VGG19 network is more adept at modeling the texture feature patterns contained within the image patches than the Resnet50 model, where the larger receptive field allows for better detection of larger objects within an image.

Figure 4.10 displays examples of misclassified images from the four-class CNN. Some observations made on these images include the fact that there are debris-like cells present in many of the images. This is because any cells that die within a colony float to the surface of the colony as explained previously. These dead cells then present themselves as debris in an image, even though the majority of the colony is still living. Another observation of note is the misclassification of differentiated colonies as Spread, and vice versa, because of the dark areas present in both of the image classes.

Most importantly, it can be seen, in comparison to the example image patches in Figure 4.3, that there are some images in the spread and dense classes that look very similar to one another. This is because these two classes display very similar morphological characteristics and are closely related biologically. For example, the third image in the third row (Spread) which is misclassified as Dense, displays larger, flatter, and protruding cell areas, especially on the periphery, which is common amongst progenitor cells. While there are other misclassifications present, the most prominent is between the Spread to the Dense class. To address these confusions, a hierarchical CNN/Triplet-net approach is implemented to filter out Dense and Spread images before using the Triplet-net model to classify these two closely related classes.

Hierarchical Classification Results

In the first stage of hierarchical classification, a three-class CNN is used to separate Debris and Differentiated images from Dense and Spread. Results of this three-class classification are shown in Table 4.6. The TPR, TNR, and F1 scores for the Debris and Differentiated class do not change noticeably, but the TPR for the combined Dense/Spread

Class/Metric (std.)	Debris	Dense/Spread	Differentiated
True Positive Rate	0.8731 (0.0088)	0.9745 (0.0021)	0.9354 (0.0091)
True Negative Rate	0.9791 (0.0015)	0.8854 (0.0078)	0.9958 (0.0006)
F1 Score	0.8902 (0.0033)	0.9706 (0.0012)	0.9148 (0.0086)

Table 4.6: Average of 5-fold Classification metrics for the Three-Class CNN

	Predicted Class		
True Class	Debris	Dense/Spread	Diff.
Debris	777	66	4
Dense/Spread	117	3531	8
Diff.	4	8	152

Table 4.7: Confusion Matrix for Three-class CNN

class increases by approximately 7% in relation to the four-class CNN (∼90% to ∼97%). Observation of the accompanying confusion matrix for this task, shown in Table 4.7, reveals fewer misclassifications of Debris as Dense or Spread, indicating that the model is able to learn a clearer boundary between these classes.

In the second stage of classification, images filtered by the three-class CNN into the combined Dense/Spread class are sent to the Triplet-net to be split into their individual classes. The results of this second stage are shown in Table 4.8 with accompanying confusion matrix in Table 4.9. Using this method, there is an increase of ∼3% from 90.44 to 93.51% for the Dense class and ∼2% from 90.92% to 92.49% for the Spread class. These results are obtained using the Triplet-net that is tested with by-hand selected anchor images. Using a student t-test, these differences are determined to be significant with a p-value of 0.0324 and 0.0360 respectively, given a significance level of 0.05. These improvements indicate that applying a hierarchical classification process including Triplet-net classification improves the

Class/Loss Config.	Dense	Spread	Average
Triplet by-hand	0.9351 (0.0059)	0.9249 (0.0039)	0.9300
Triplet auto (k=11)	0.9318 (0.0094)	0.9182 (0.0095)	0.9250
Triplet auto (k=2)	0.9005 (0.0322)	0.9327 (0.0125)	0.9166
Triplet LDA	0.9465 (0.0124)	0.9243 (0.0073)	**0.9354**
Triplet corr.	0.9377 (0.0107)	0.9221 (0.0078)	0.9299
Triplet random	0.9138 (0.0092)	0.9309 (0.0021)	0.9224
Triplet softmax	0.9067 (0.0021)	0.9384 (0.0062)	0.9226
softmax	0.8459 (0.0222)	0.9635 (0.0060)	0.9047
MIL [81]	0.8203 (0.0153)	0.8966 (0.0084)	0.8584
MTM [92]	0.2227 (0.0241)	0.7174 (0.0013)	0.4701

Table 4.8: Two-class Results (True Positive Rate (std.))

	Predicted Class	
True Class	Dense	Spread
Dense	915	65
Spread	211	2413

Table 4.9: Confusion Matrix for Triplet CNN Classification

ability of the model to distinguish between these closely related classes. The Triplet-net provides the model with the ability to make more informed decisions by using the anchor images to make distance measurements between learned features, instead of only decision level classification based on the learned distribution.

4.5 Results and Discussion

4.5.1 Comparison of Results

Comparison with Triplet-net Configurations

Additionally, these results were compared to those of both a straightforward, two-class classification CNN (as in [5]), as well as a Triplet-net configuration using an aggre-

gate loss function that includes Triplet-loss and a softmax classification loss (similar to [45][88][90]), which avoids the use of anchor images in the testing phase of network evaluation. Some related methods were not used for comparison because the triplet-net implementation or novel loss functions is not applicable to the task performed in this work [84][86][87][89]. The results of these comparisons are shown in Table 4.8 and are used to determine the effectiveness of the Triplet-net configuration in comparison to other methods of classification. It is observed that, on average, the Triplet-net classification results are better than those of the other network configurations, on average 0.93 vs. 0.92 and 0.90 for the Triplet-net, Triplet-softmax, and softmax respectively. This is due in part to the use of thousands of Triplet pairs for training, in comparison to a straightforward CNN, and to the use of anchor images in classification, as well as the reduced effect of Triplet loss, in comparison to the network trained with the Triplet-softmax aggregate loss function.

Visualization of Learned Features

As previously noted, Triplet-net is able to use contrastive comparison to learn a more accurate representation of the image features. As a visualization of this effect, Figure 4.11 displays the t-distributed Stochastic Neighbor Embedding (t-SNE) plots of learned feature embeddings for Dense and Spread classes as modeled by the Triplet-net and CNN architectures [104][105]. These plots are obtained by gathering the feature embeddings for the Triplet-net and CNN networks before their respective classification layers, performing principal component analysis (PCA) dimensionality reduction from 512 features to 50, and then performing t-SNE reduction (perplexity: 50, iterations: 1000) down to three features, which represent the x, y, and z axes of the plots. Observation of these charts reveals

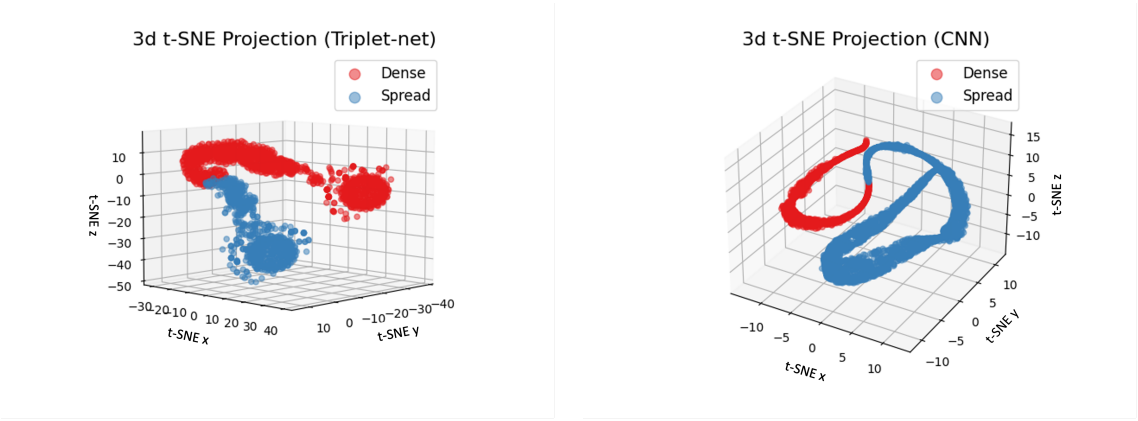

Figure 4.11: t-SNE plots for triplet-net (top) and standard CNN (bottom) configurations. x, y, and z axes are representative of the three low-dimension t-SNE features.

relationships between class-wise image features in low-dimensional space, including the two larger cluster centers at the tails of feature distribution of the triplet-net, whereas those for the CNN are much longer and evenly distributed. This may be indicative of the ability of the triplet-net to more accurately group similar features of the two classes. These two plots show some similarities such as general topology and shape, including intersection between image features, which corroborates the assertion that these classes share close relationship.

Effect of Anchor Selection on Network Performance

As mentioned previously, the subjective nature of anchor selection is a limitation of the Triplet-net method, especially for biological datasets. By-hand image selection requires a skilled worker to determine microscopy images that are most representative amongst a dataset containing thousands of samples. To simplify and standardize this process, several automated anchor image selection methods are employed to leverage morphological and texture features for anchor selection. The effect of this anchor selection method on network

performance is determined by training separate network configurations using automatically selected anchor images using different selection methods and comparing these results to the configuration using hand selected anchors.

Multiple methods of feature clustering and classification are used in order to determine the effect of the selection method on network performance. These methods have various implications on the relationship of images in the dataset, as described in Section 4.4.4. Table 4.8 displays all of the classification results of each network trained with different anchor images. Of all the selection methods, anchor images chosen using the Linear Discriminant Analysis method show the highest accuracy, on the average, across both of the image classes classified using Triplet-net. This method of anchor selection produces the best results because the goal of the LDA algorithm is able to maximize the ratio of $\frac{|S_B|}{|S_W|}$ via Fisher's discriminant. Maximizing this value ensures that the most representative images of each class are also those that are furthest from the opposite class. These results outperform those of the anchor images selected by hand and demonstrate the ability of automated anchor selection methods in determining representative images across the whole dataset.

From Table 4.8 it is determined using student t-test that there is no statistically significant difference between the Triplet-net results using anchor images chosen by-hand (Triplet w/ by-hand) or using the automated selection method (Triplet w/ auto). Furthermore, both of these results outperform the alternative method of avoiding anchor images via softmax classification results (Triplet + softmax). This illustrates the effectiveness of the automated anchor selection method in determining representative images and helps improve the reliability of the model in terms of repeatability and generalizability. This method can

potentially be adapted to new datasets and removes user error and bias in image anchor selection.

Comparison with State-of-the-art Related Work

To assess the validity of the proposed triplet-net method, two existing approaches are trained to discriminate between the Dense and Spread image classes. These methods are specifically chosen as comparisons because they represent the state-of-the-art in terms of applicable approaches towards the proposed task, more specifically multiple-instance learning and multi-template matching. As mentioned in Section 4.3, Campenella et al. [81] use multiple instance learning (MIL) to classify histopathology images of various cancer types and combine CNN and RNN to make slide-level class predictions. MIL is a learning strategy that uses positive instances of individual classes to make predictions over a larger set of images. In this case, positively detected image patches within very large slides are used to predict clinical outcomes of cancer pathology. This method is comprised of three separate training steps; inference, MIL training, and RNN aggregation. First, the network is used to determine high-probability patches within an image, which are ranked and given to the network again for feature learning during the second step. Finally, the RNN is used to aggregate image features from multiple patches within a colony region of interest in order to make a more globally informed prediction.

This method can be applied to the dataset in this work because image patches are being used to make predictions about colony level classification. This comparison highlights the ability of multi-step, combinatorial deep learning models to discriminate between visually similar morphological classes. In practice, this method does not perform as well the

proposed method, in terms of TRP (93.54 % vs. 85.84% on average, Table 4.8). It is difficult to determine which stages of the MIL model would contribute to these misclassifications, however, it does speak to the utility of the triplet-net model in terms of generalizability and debiasing.

Alternatively, Thomas et al. [92] use multi-template matching (MTM) via normalized cross-correlation coefficient to detect zebra fish embryos in microscopy images. They extend the standard template matching algorithm to consider several templates simultaneously within an image while controlling parameters such as overlap and confidence threshold. MTM is relevant to the proposed method because it represents a contemporary implementation of standard template matching methods for biological image data. However, MTM is limited in its ability to recognize texture features within similar images. This is apparent when looking at the TPR results in Table 4.5, where MTM is able to more accurately classify the spread class (71.74 %), which contains larger, more simple feature patterns. On the other hand, the dense class contains many fine-grained textures, which are not able to be recognized using template matching directly, resulting in low classification accuracy (22.27 %). In contrast, the feature embeddings learned by the triplet-net are much more robust such that comparison with the same templates at the embedding level results in a much higher true positive classification rate.

These two methods represent the state-of-the art in terms of deep learning and classical image recognition techniques and serve as ample comparison for methods that are able to perform two-class classification of images. The triplet-net method proposed in this work outperforms the aforementioned related approaches, and results in unbiased, and

robust classification results for the very closely related Dense and Spread classes.

Effect of Number of Anchor Images on Classification Performance

The effect of the number of anchors used to test Triplet-net configurations is determined empirically by testing the Triplet-net with an increasing number of anchors images. Figure 4.12 displays the classification accuracy of Triplet-net tested with between 1-11 anchors and shows that there is an increase in accuracy until 5 anchor images are used, after which the accuracy does not increase. These results are compiled by performing Triplet-net testing with anchor each image individually as well as exhaustive combinations of images totaling the values for the indicated bins. The variance of testing accuracy is shown as error bars for each data point. The standard deviations for the 1, 3, and 5 anchor testing configurations are 0.36, 0.27 and 1.9×10^{-4} respectively, and 0.0 for the 7, 9, and 11 anchor image configurations. These findings indicate that a greater number of anchor images reduces variability in testing accuracy. While all 11 anchor images were used in the testing of the model configurations in this work, it may not be necessary to remove the extra images from the training dataset given the trade-off between number of anchor images and classification accuracy, noting that the variance goes to zero using at least 7 anchor images.

Overall, this hierarchical approach improves classification performance in comparison to a four-class CNN. The Triplet-net CNN is able to learn more fine-grained features of the Dense and Spread classes, which it can use to distinguish between the visually similar image patches. By compartmentalizing the classification task according to biological hierarchy, the model can learn more distinct boundaries between closely related classes in order

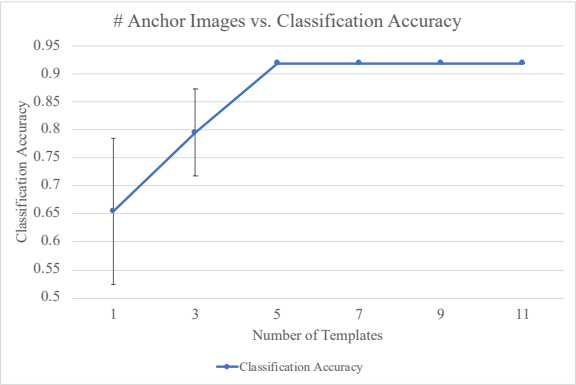

Figure 4.12: Average true positive rate for Triplet-network given different number of Anchor images used during testing. In this work 11 anchor images are used during Triplet-net testing phase. However, the testing accuracy does not improve by using more than 5 anchor images, suggesting that it is not necessary to use more images as testing anchors.

to make more informed decisions. In turn, this method can be employed to distinguish between these classes in images containing multiple, contiguous cell types such that, image attributes, such as colony area-per-class, can be quantified in a non-invasive manner using only morphological features.

4.5.2 Bioengineering Implications

There are several potential uses and implication of this work on experimental analysis and industrial scale bio-therapeutics. Most immediately it can be used to determine the effects of environmental factors on cellular behavior, such as the effect of Nicotine as a potential therapeutic candidate for Huntington's disease in this work. Observations of neuronal behavior under these conditions can help to elucidate the functional relationship between cellular morphology and disease manifestation. Similarly, it can be employed to track and quantify cellular events such as differentiation, in order to build a model of cellular growth and change.

In subsequent studies, this model can be expanded to automatically quantify experimental endpoints in a dataset-agnostic manner and to help develop experimental assays involving microscopy data acquisition. For example, by encompassing multiple imaging modalities or by incorporating bio-molecular data. On a larger scale, this network has the potential to perform high-throughput quality control for stem cell therapeutics and clinical trials with high specificity, a crucial aspect of clinical success.

4.6 Conclusions

This work leverages Triplet-net CNN training to improve the accuracy of a four-class classification task involving stem cell microscopy images. In comparison to a traditional 4-class CNN, this hierarchical method improves classification accuracy by approximately 3% in terms of true positive rate, a statistically significant increase, and 2% decrease in terms of false alarms. This approach implements two stages of hierarchical classification to first filter images into three classes, with a combined class for two visually similar inputs, and then sort those images into their individual classes using a Triplet-net. Furthermore, the anchor image selection process, which is highly subjective, is standardized and automated using feature clustering to determine representative images. The model is used to overcome misclassifications as a result of morphological similarities between stem cell classes undergoing differentiation as well as solve data imbalance issues associated with biological datasets.

In a real-world setting, this approach can be used in biological experimentation to determine the effects of nicotine on the behavior of diseased stem cells with respect to the

classification task. For example, by exploiting the observation of debris area within colonies to determine health status. This is paramount to improving the reliability of experimental quantification and increasing the throughput of analysis in order to keep up with high-throughput imaging techniques. Future work includes using temporal information to model dynamic colony features, as well as integrating bio-molecular markers for validation and improved performance, which currently limits the information available for model learning to morphological features.

Overall, this work represents an improvement in performing non-invasive biological quantification using only morphological image features. The method presented here allows for the multi-label classification of colony images with contiguous, overlapping classes. The modeling of fine-grained features between visually similar classes gives merit to the hypothesis that these cells are biologically distinct, up/down stream precursors/progenitors of one another. The triplet-net model in this work removes sources of error caused by bias inherent in biological data acquisition and out-performs existing state-of-the-art approaches by exploiting class relationships and biological domain knowledge. The findings presented in this work highlight the importance of designing deep learning models using biological constraints and synergistic design of biological experiments.

Chapter 5

Iterative Pseudo Balancing for Stem Cell Microscopy Image Classification

5.1 abstract

Many critical issues arise when training deep neural networks using limited biological datasets. These include overfitting, exploding/vanishing gradients and other inefficiencies which are exacerbated by class imbalances and can affect the overall accuracy of a model. There is a need to develop semi-supervised models that can reduce the need for large, balanced, manually annotated datasets so that researchers can easily employ neural networks for experimental analysis. In this work, *Iterative Pseudo Balancing (IPB)* is introduced to classify stem cell microscopy images while performing on the fly dataset

balancing using a student-teacher meta-pseudo-label framework. In addition, multi-scale patches of multi-label images are incorporated into the network training to provide previously inaccessible image features with both local and global information for effective and efficient learning. The combination of these inputs is shown to increase the classification accuracy of the proposed deep neural network by 3% over baseline, which is determined to be statistically significant. This work represents a novel use of pseudo-labeling for data limited settings, which are common in biological image datasets, and highlights the importance of the exhaustive use of available image features for improving performance of semi-supervised networks. The proposed methods can be used to reduce the need for expensive manual dataset annotation and in turn accelerate the pace of scientific research involving non-invasive cellular imaging.

5.2 Introduction

Stem cell biology is a promising field of study that has applications in the areas of regenerative medicine and disease modeling [106, 107]. In-vitro experimentation is paramount to determining the underlying mechanisms of cellular growth and differentiation that contribute to a more comprehensive understanding of stem cell biology. However, one bottleneck in experimentation is the reliable and robust analysis of non-invasive microscopy imaging used to observe cellular changes. This work proposes to use deep learning algorithms to automatically analyze microscopy images containing multiple cell types with dynamic morphological structure. Specifically, problems arising from highly imbalanced datasets are addressed here using pseudo-labeling to allow for the use of small, imbalanced

datasets for training data-expensive neural networks to characterize stem cell health and differentiation status. The models presented in this work will allow for more efficacious implementation of deep learning in research based settings to reveal key insights into stem cell biology for use in a variety of applications.

Stem cell differentiation is a delicate in-vivo process that orchestrates embryonic changes from naive pluripotency to somatic lineage commitment in all living organisms [7]. These cells are most susceptible to harm at the earliest stages of growth and development, when the small number of cells that make up the embryo are easily affected by environmental toxicants and mutagens [15]. The biochemical understanding of the underlying mechanisms of these processes has been harnessed towards technologies that are aimed at regenerative medicine and cellular manipulation such as induced pluripotent stem cells (iPSC), which are a useful tool for researchers to understand the developmental differentiation process including models of disease [14, 12].

One such disease, Huntington's disease (HD), is a neuro-degenerative disorder that affects the HTT gene in the human brain [58]. This mutation causes a repeat expansion of the CAG codon at the terminus of the Huntingtin protein. This protein is instrumental in neuronal maturation and migration, and the functional gain incurred by HD expression prompts these processes to occur at a much higher rate. Like other proteopathic disorders (e.g. Parkinson's, ALS, etc.), HD manifests itself in muscular skeletal disfunction and rapid cognitive decline [108, 57, 109]. These symptoms display themselves at later stages of life, and there is evidence to suggest that nicotine has a neuroprotective effect that can slow disease progression [73, 110]. Given this, and the fact that HD has been shown to

affect human neurodevelopment in both humans and mice by causing neural progenitor cells to mature at a faster rate, it may indicate that nicotine could have the same effect on developing stem cells [108].

Live cell microscopy of stem cell disease models is crucial to the research efforts that have made these important discoveries. These non-invasive methods allow scientists to observe cellular behavior in response to experimental stimuli, and predict outcomes based solely on morphological patterns. Video-bioinformatics (VBI), which is defined as "the automated processing, analysis, understanding, data mining, visualization, query-based retrieval/storage of biological spatiotemporal events/data and knowledge extracted from videos obtained with spatial resolution varying from nanometer to meter of scale and temporal resolution varying from seconds to days and months," has led to the advancement of programs aimed at quantification of stem cell events [16]. Such programs include those that use traditional computer vision, pattern recognition, and machine learning classification techniques [4, 20, 111]. Others have recently adopted deep learning, which combines the feature extraction and classification modules, learns a more robust representation of input images and significantly out-performs hand-designed methods in a majority of VBI tasks [78, 1, 31, 5, 81].

The unique circumstances of biological experimentation and data collection (i.e., cost, time-constraints, uncertainty in experimental outcomes, and lack of image data) reveal some drawbacks to deep learning. Factors such as class imbalances, contiguous boundaries, multi-label images, anomalies, lack of data, and the prerequisite of manual annotations hinder the ability of researchers to easily employ deep learning models for their analytical

pipeline. These limitations are a major hurdle in training neural networks using biological images.

For example, in the case of stem cell microscopy, many multi-label images are collected that contain more than one morphological class in a single image. It is very difficult to use multi-label images for supervised training because they cannot be accurately annotated without invasive biomarker validation. Unlike natural image datasets, the task of biological image annotation must be performed by a domain expert or someone trained to specifically analyze a particular dataset. Even with this requirement satisfied, reliable annotations cannot always be guaranteed given the high degree of variability within biological images.

Even when manual annotations have been provided to every image, a subsequent problem is the innate imbalances in the dataset class distribution. In stem cell biology, these imbalances arise from the pluripotency of cells before undergoing differentiation, meaning that they have the ability to change into any mature cell type. In-vivo, these changes are orchestrated by a very specific and well-defined cascade of signals from the developing embryo [112]. However, in-vitro they are subject to a measure of uncertainty that results in an imbalance in proportions of the observed image class samples[13]. This randomness, coupled with the downstream differentiation of cells towards somatic lineages causes variation in the number of cells at any given stage of development during the experimental growth cycle.

These drawbacks can be overcome by leveraging the power of semi-supervised learning algorithms rather than those trained end-to-end in a fully supervised manner.

These algorithms allow the network to estimate features from an imbalanced, unlabeled dataset and fine-tune these features via reinforcement from a smaller, balanced, labeled subset. In a practical setting, this allows researchers to hand label a smaller portion of their dataset and train a network on both labeled and unlabeled data without having to perform extensive manual annotations. This saves time and resources and improves analytical workflows involving deep learning.

Therefore, this paper introduces Iterative Pseudo Balancing (IPB), which uses meta-pseudo-labeling to balance a dataset of biological images on-the-fly during model training and improve the performance of a deep neural network for the task of stem cell colony classification. The outline of the paper is as follows: First, related works are discussed and compared to the proposed method, and the unique contributions of the paper are clearly stated along with their significance. Second, an overview of the technical approach is presented, along with in-depth explanation of the novel aspects of the network architecture, algorithm, and experimental configurations. Finally, the results of these experiments are summarized and interpreted, and inferences are drawn as to the reasons behind the observed patterns. Ultimately, it is determined that the method presented in this work improves over previous works by exploiting domain knowledge for network training, and advances the state-of-the-art by solving the problem of class imbalances in biological image datasets.

5.2.1 Related Work and Contributions

Semi-supervised deep learning has recently been leveraged for image classification tasks with a popular method being the contrastive learning networks. Contrastive learning refers to training a neural network by comparing similar and dissimilar image samples and

updating the feature map in relation to a comparative loss function [113]. For example, SimCLR, proposed by Chen et al. [114] uses contrastive learning to map input features based on the similarities and differences between individual images. At each iteration, a single image is considered a positive instance while all other images are considered negative instances. The network learns a general view of the input features, and then it is refined via supervised fine-tuning to incorporate class information. This is an effective self-supervised method, however, there is no way to account for class imbalances in contrastive learning without prior information, which limits the ability of the network to learn unbiased features. These algorithms also require an extremely large dataset for effective learning, which is counterproductive for a task involving limited datasets.

Another work by Chuang et al. [115] called Debiased Contrastive Learning (DCL) attempts to improve upon the contrastive scheme by removing some randomness from the sample selection process. DCL compensates for sampling bias by estimating the probability distributions for the negative and positive classes. However, they assume that these distributions are uniform, which is often not the case in real world, biological settings.

Similarly, Li et al. [116] introduce Contrastive Clustering, which uses two output layers to extract features at the instance and cluster levels. They try to approximate the feature space more closely by updating the network with respect to a joint contrastive loss function that combinies these two feature modules. They outperform various state-of-the-art clustering methods, but report that their method is highly dependent on data bias and more research is needed to improve model robustness to the level of real-world applications such as health care.

Previous studies have attempted to solve the problems associated with biological datasets, including limited data and imbalanced class distributions, using semi-supervised, self-supervised and contrastive deep learning methods. One such work by Murphy et. al. [117] uses a SimCLR model to extract feature embeddings from a dataset of immunohistochemicstry (IHC) images and combine them with protein data to predict image biomarkers in a self-supervised manner. They implement a hard negative sampling scheme to counterbalance inherent dataset biases and outperform baseline methods of transcriptomic classification but fail to beat fully supervised training methods aimed at histopathology [118]. They note that the variability within IHC images and between image samples aids in the augmentation schemes necessary for effective learning using contrastive algorithms and theorize that multi-scale inputs would help to improve their training. However, their method utilizes concurrent genetic information in the form of scRNA sequencing data, whereas the method in this work seeks to determine class features based only on morphological information from non-invasive live-cell imaging.

Another work by Liu et al. [119] aims to classify histopathology images using a contrastive network that combines the SimCLR framework with a triplet-net configuration to simultaneously minimize intra-class variance and maximize interclass variance. Their model, SimTriplet, avoids performing negative image sampling by assuming that adjacent patches within a large scale image contain features from the same image class, while patches from other image regions contain pertinent negative context. Their model overcomes the computational expense of the SimCLR model to improve overall accuracy for their classification task, however, they perform pre-balancing of the dataset to compensate for dataset

imbalances, and only use a single-scale, down-sized image patch to improve training efficiency, as opposed to a multi-scale input.

Other similar patch based methods exist such as Visual Transformers (ViT), which break images down into patches to perform attention with state-of-the-art classification results, but require large datasets for effective learning (15-100M images) [120]. This requirement has a similar pitfall to contrastive learning, and can affect the results of networks trained on smaller datasets. A network called DeepFA, by Benato et al.[121], attempts to solve this problem by using a student-teacher network that combines a 2D semi-supervised forest classifier with a meta-pseudo-label (MPL) module to learn image features from a small data subset on a variety of datasets. They use their method to solve the problem of low data availability, but use random sampling in their training and note dataset imbalances as a limitation of their work.

Given the drawbacks of the previously described methods, the approach outlined in this work aims to balance and classify a stem cell microscopy dataset using a semi-supervised student-teacher framework with multi-scale inputs from multi-label images. To the best of our knowledge, no other similar works have been published that address biological image classification using meta-pseudo-labeling. The specific contributions of this paper are as follows:

- Introduces Iterative Pseudo Balancing to classify stem cell microscopy images using semi-supervised learning and address dataset limitations surrounding manual annotation and class imbalances; improvements over state-of-the-art will allow researchers to accelerate their experimental analysis using deep learning without the need for large labeled datasets

- Performs on-the-fly dataset balancing via pseudo-label-resampling at the image patch level. This is the first use of psuedo-labeling for dataset balancing, which allows for neural network training with unlabeled datasets. The previous MPL algorithm [122], did not account for imbalanced datasets, which is detrimental to model learning

- Incorporates previously inaccessible image features using patches of multi-label images to provide local and global features via multi-scale inputs. It is shown here empirically that the combination of these features is necessary for the greatest improvement in classification accuracy and improves over the single scale inputs of state-of-the-art related methods

5.3 Technical Approach

5.3.1 Iterative Pseudo Balancing Framework

Biological datasets present unique circumstances for image feature modeling and classification. Unlike the standardized natural image benchmark datasets such as ImageNet[23] or CIFAR10 [34], medical and biological images often require extensive curation and preprocessing as well as special considerations for network architecture and training procedures.

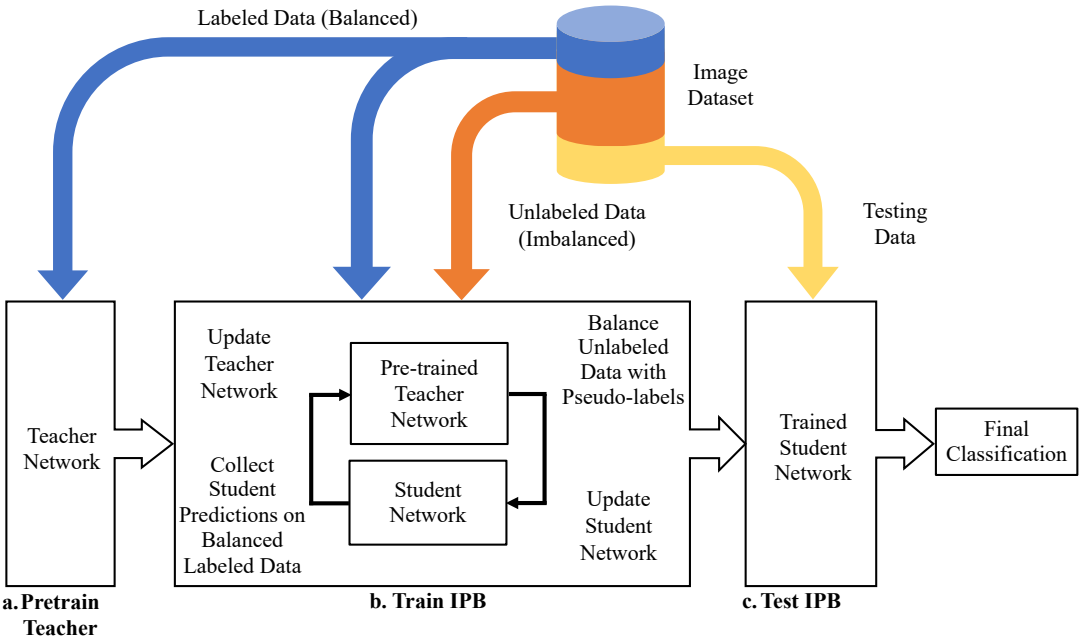

Figure 5.1: The overall diagram for the Iterative Pseudo-Balancing framework. (Top row) The dataset is divided into three parts, two smaller labeled subsets for training the teacher and testing the student, and a larger, unlabeled dataset for training the student network using iterative pseudo-balancing. (Bottom row) a. the teacher network is pre-trained on the balanced, labeled dataset. b. the IPB algorithm uses the pseudo-labels from the pre-trained teacher to resample and balance the unlabeled dataset during each epoch for training the student network. The teacher is then updated in relation to the classification performance of the students predictions on the labeled data. This process is repeated until the network converges. c. The trained student network is validated on the testing dataset. Details of this procedure are outlined in the section below.

For example, regions-of-interest within microscopy images generally contain high variability in terms of local entropy and fine-grained patterns. Furthermore, manual annotation across an entire dataset for use in neural network training poses practical issues in terms of time, and analytical redundancy caused by having to pre-sort every image in the dataset before training the neural network. In other words, it defeats the purpose of training a machine learning algorithm for biological image classification if manual annotation must be performed on a large dataset to train, test, and validate the initial model.

Another very important consideration is the effect of class imbalances on semi-supervised learning. The meta-pseudo-labels [122] algorithm utilized in this work suffers from overfitting as a result of confirmation bias when trained on imbalanced datasets. This causes the model to devalue the features of the least prevalent classes in its classification decision. Therefore, it is necessary to provide the semi-supervised model with a balanced view of the dataset classes to avoid overfitting on the most prevalent class. In this paper, Iterative Psuedo Balancing (IPB) is introduced to address these challenges by using semi-supervised pseudo-labeling to balance image classes. Figure 5.1 describes the proposed approach for the IPB framework. The numbered training and testing steps are as follows:

a. Training Initialization

1. The dataset is split by taking a portion of the images in each class for training the student using pseudo-labeled images (U), pre-training and updating the teacher network (L) and testing the student network (T).
2. Training is performed over the course of N epochs, for which a single epoch is completed when the student network has seen every image in the unlabeled dataset (U).
3. The HRNet configuration is used for both the teacher and student networks with the training specifications as shown in Figure 5.3.
4. The weights of both the teacher and student networks are initialized using Kaiming initialization [123].

b. Pre-train Teacher Network

5. The teacher network is pre-trained, in a fully supervised manner, using a small, balanced, labeled subset of the dataset (L). Pre-training allows the teacher network to provide informed pseudo-labels to the unlabeled dataset during the iterative pseudo balancing phase, however, the teacher is also updated during the IPB phase in relation to the students performance on the labeled dataset.

c. Train IPB

6. The teacher network (initially pre-trained) is used to provide pseudo-labels to random image patches (one from each image) from the unlabeled dataset (U). The pseudo-labels are then used to balanced the imbalanced dataset using resampling from a multinomial distribution in relation to the weighted class proportions as determined by the teachers pseudo-labels.

7. The balanced pseudo-labeled dataset is used to update the student network by collecting the students predictions on the pseudo-labeled images patches. The weights of the student network are updated in relation to the cross-entropy loss between the students predictions and the image pseudo-labels from the teacher (see Equation (5.1)).

8. To update the teacher, the predictions from the updated student network on the labeled images (L) are collected and the cross-entropy loss between the students predictions on the labeled dataset and the actual class labels is used to update the teacher (see Equation (5.2)). This allows the teacher network to learn more

robust pseudo-labels with each training epoch in order to provide the student with more accurate pseudo-labels.

9. Steps (6-8), are repeated for N epochs until both the teacher and student networks converge, as monitored by the cross-entropy loss values from each network, as shown in Figure 5.2. At the end of training, the student network with the highest classification accuracy over the course of training is taken as the final network for testing.

d. Test IPB

10. The trained student network is evaluated with the remaining testing data (T). One patch from each image in the testing dataset is provided to the network to perform image level testing. Network predictions for each image are collected and compared to the image labels for the testing dataset to perform final classification.

11. The final classification is the maximum value of the softmax probability across the four morphological classes for every image in the testing dataset.

5.3.2 Meta Pseudo Labels Algorithm

As previously discussed, self-supervised and semi-supervised methods present a unique opportunity to avoid expensive pre-processing, curation, and manual annotation in training neural networks. However, these contrastive learning methods do not adequately account for dataset imbalances and fail to incorporate domain knowledge to guide model learning. Therefore, there is a need to address these issues using semi-supervised learning

so that researchers can more effectively employ deep learning to standardize and accelerate their experimental analysis. To accomplish this, a meta-pseudo-labeling (MPL)[122] algorithm is used as a basis for the proposed framework. MPL uses a student-teacher network configuration to learn features from unlabeled data, where a small amount of labeled data is used to first pre-train the teacher network, which then provides pseudo-labels for the other portion of unlabeled data at every iteration that is subsequently used to update the student network. The teacher network is updated in relation to the loss associated with classifying the labeled data using the student network as well as with a contrastive loss value inspired by Unsupervised Data Augmentation (UDA) [124].

$$\theta'_S = \theta_S - \eta_S \nabla_{\theta_S} \mathcal{L}_S(\theta_T(x_u), \theta_S(x_u)) \tag{5.1}$$

$$\theta'_T = \theta_T - \eta_T \nabla_{\theta_T} \mathcal{L}_T(x_l, \theta'_S(x_l)) \tag{5.2}$$

The loss functions for the teacher and student networks shown below highlight the relationship between the two networks during training. Equation (5.1) is the learning equation to update the parameters of the student network, where the updated student weights, θ'_S, are calculated from the initial network weights, θ_S using the gradient (∇_{θ_S}) of the cross-entropy loss, $\mathcal{L}_S$, between the pseudo-labels provided by the teacher network for the unlabeled images $\theta_T(x_u)$ (where (x_u) is the unlabeled data) and the predictions of the student network on the pseudo-labeled images, $\theta_S(x_u)$, where η_S is the learning rate of the student network. Equation (5.2) is the learning equation for updating the parameters of the

the teacher network, θ_T, using the gradient (∇_{θ_T}) of the cross-entropy loss, $\mathcal{L}_l$, between the predictions of the updated student network on the labeled images, ${\theta'}_S(x_l)$, and the actual class labels, x_l, to obtain the new parameters of the teacher network ${\theta'}_T$, where the learning rate of the teacher is η_T. In this way, the teacher is iteratively learning from the student network and vice versa, such that the teacher can also learn more robust psuedo-labels for the unlabeled data.

In a practical research setting, the MPL algorithm allows for a novel microscopy image dataset to be collected, partially annotated, and for the remainder of the raw data to be used as unlabeled input to train the student network in a semi-supervised manner. Consequently, this makes it possible to utilize patches of multi-label images in the training set without having to provide patch level semantic labels. For example, in the case of an experimental protocol for which multiple experimental folds are conducted, the researcher could use a single fold for training and testing the MPL network, and the other folds for cross-validation. In this scenario, the researcher could manually annotate a small portion of one fold for pre-training and testing the teacher network, use a larger, unlabeled, portion of that fold to train the student via MPL learning, and analyze the other folds using the trained network. Subsequent experiments with similar visual classes could then be analyzed in real time using the trained and tested student network without having to annotate more data, and network fine-tuning can be performed to incorporate new data classes. This setup overcomes the need for researchers to annotate the entire dataset for fully supervised learning, greatly improving the efficiency of the image analysis pipeline.

As previously stated the MPL algorithm does not inherently account for class imbalances within a dataset, which can lead to model bias and overfitting. Furthermore, none of the contrastive methods reviewed here include a multi-scale input, which limits the available features for the network to learn. Multi-scale inputs have been used for biological image classification to incorporate large scale image features while still allowing images to be classified by the fine-grained texture features of cellular image classes [26]. Therefore, the approach outline in this work, Iterative Pseudo Balancing (IPB), utilizes the pseudo-labels estimated by the MPL algorithm to iteratively resample a dataset of image patches such that for each epoch, the network is provided with a balanced dataset, helping to improve model learning. Furthermore, both multi-scale and multi-label inputs are used to improve feature extraction and classification of the network.

Training Procedures and Architectures

Multiple network configurations are tested using the IPB algorithm to compare the effect of network architecture on stem cell classification accuracy. Different networks have certain advantages and drawbacks depending on the dataset because of the effect of receptive field on image feature mapping [103]. Specific parameter values such as kernel size, convolutional stride and overlap, and network depth and width contribute to the modeling of features and are often best suited for specific tasks. For example, VGG [94] is optimal for modeling fine-grained features like those found in biological datasets, whereas ResNet [79] is better for detection of larger objects and regions-of-interest (ROI). In this paper, the VGG19 architecture is used as a baseline configuration, and has been shown previously to produce superior results in comparison to ResNet on stem cell images [2].

The High Resolution Network (HRNet) [125] is also tested in this work as an example of a more recent network architecture that overcomes the loss of information in low level convolutions by combining network features in parallel to conserve high-level features in deeper layers. HRNet fuses layers along parallel branches in order to conserve spatial features at multiple resolutions, as opposed to convolutional networks that combine layers in series, which results in a loss of high-resolution information at deeper layers. In this way, HRNet is able to more effectively represent high and low level image features with a similar number of overall parameters and computational cost as other neural network architectures.

All network configurations are trained from scratch using 10-fold cross validation, with an 80:10:10, Unlabeled Training:Labeled Training:Testing dataset split. The 10% labeled data is chosen as a standard benchmark for semi-supervised learning methods and is used here to simulate a limited data setting [122]. The balanced, labeled training data subset, L, is used to pre-train the teacher network, as well as update the teacher during IPB training based on the students predictions over the labeled images. The unlabeled training data, U is used to train the student network with the pseudo-labeled patches from the pre-trained teacher. The labeled testing dataset, T, is used to evalutate the network at the end if IPB training, and is unseen by either of the networks during any of the training steps.

The HRNet used for both the teacher and student networks contains 10 convolutional layers and 5 fully connected layers for a total of 15 layers. All convolutional layers use 3×3 filter kernels with stride of 1 pixels and padding of 1 pixel. Each convolutional layer is followed by a batch normalization function and ReLU activation. Down-sampling layers

use 3×3 filter kernels with stride of 2 pixels and padding of 1 pixel. The number of filter channels in each layer is given in Figure 5.3. A batch size of 32 is used during pre-training and training of the IPB algorithm. Both the teacher and student networks are initialized using Kaiming Initialization [123] before beginning any of the training steps.

Hyperparameters for the stochastic gradient descent optimizer were determined empirically and include a learning rate of 0.005, weight decay of 0.0001, and momentum of 0.9, as well as the number of training epochs, which is the epoch at which training stabilizes as determined by the cross entropy loss. The teacher network is pre-trained on the small labeled dataset for 200 epochs and the IPB network is trained for of 200 epochs, where one epoch is determined when the student sees every pseudo-labeled image of the unlabeled dataset. Dataset augmentations (described in the following section) are performed on the training dataset to increase image variability, and ensure spatial invariance. When IPB training is complete, the student network with the highest accuracy is taken for performing network evaluation using the labeled testing dataset. Figure 5.2 displays the loss function for the student and teacher networks. The IPB training algorithm performs 10 epochs for warm-up of the student network, as evident by the plateau present at the beginning of the loss curve for the student network.

5.3.3 Data Pre-processing

Cell Colony Detection

Before training, several data pre-processing steps are performed to reduce irrelevant information from input images. Firstly, the raw microscope images measure 2908 ×

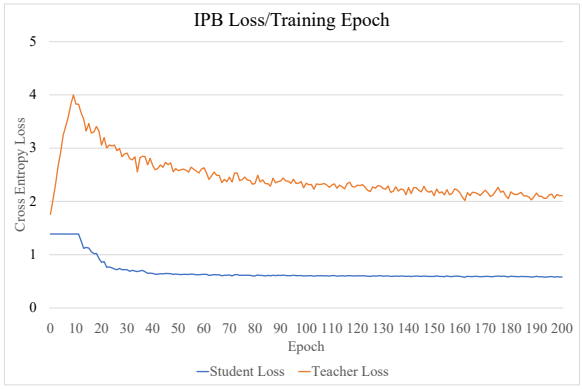

Figure 5.2: Training loss curves for the teacher and student networks, averaged over multiple runs. At the beginning of the training, the teacher loss increases while the student is still in the warm-up phase, where the learning rate is kept low to allow for the student to catch up to the teacher. After this phase, the teacher and student losses begin to go down, and stabilize over the course of training. As training progresses, the student network producing the best classification accuracy is taken as the network used for evaluation.

2908 pixels and contain several cellular colonies within a single image. To remove background area, these colony ROIs are extracted from the image using a morphological segmentation scheme that includes the following steps (with specific parameters and OpenCV (Version 4.5.5) functions included): 1. Gaussian blurring (cv2.GaussianBlur, kernel size 3×3), 2. entropy filtering (skimage.filters.rank.entropy, disk filter size 3), 3. binarization via Otsu thresholding (skimage.filters.threshold_otsu), 4. morphological opening (skimage.morphology.opening, disk filter size 3), 5. hole filling (scipy.ndimage.morphology.binary_fill_holes), 6. small object removal (skimage.morphology.remove_small_objects with filter size of 2000 pixels).

Bounding boxes containing these binarized areas are cropped out of the raw images and used to build a dataset of single colony images containing either one of the four individual morphological classes (Dense, Differentiated, Spread, Debris) or multi-label images that contain more than one morphological class. The binary maps are also used as

a boundary from which patches are taken within the borders of the cell colony, such that every image contains relevant class information. During training, random image patches are selected from the image within the colony ROI. An example of a binary map for a gray scale input image can be found in Figure 5.5.

Multi-scale Input

There is an important distinction to make when referring to scale and resolution in terms of optical microscopy and image properties. For the optical light microscope used to capture the image dataset in this work, the term resolution refers to the smallest distance between two discernible points of light captured within an image, whereas scale is used to determine the size of objects that can be observed within the image. For images captured using a specific objective magnification (in this case 10x), the resolution of the image is a fixed value that is related to the spacial distance represented by a given number of pixels, and scale is related to the number of pixels comprising an image in a given region-of-interest. These two terms can sometimes be used interchangeably when discussing images because they can both have similar effects on image output (i.e. changing image scale can also affect the resolution of the image). For the purposes of this paper, the term scale is taken to mean the size of the input image at a given resolution, and therefore multi-scale refers to taking multiple size patches from the original image with a fixed resolution.

Using multi-scale inputs can have several advantages over a single-scale because of the nature of deep feature extraction, including image down-sampling steps performed by the neural network which result in low-level feature representations of the input image, where some fine grained features can be lost in feature space. Conversely, when providing

multiple image scales, features at multiple input levels can be provided to the network for training, which allows for modeling of local and global features separately. For example, Christiansen et al. perform in-silico labeling of histopathology images by using multi-scale inputs to convert immunohistopathological images to fluorescent staining without the use of fluorescent microscopy [26]. They successfully segment various cellular structures using a multi-head output, but their network is very large and extremely expensive computationally, making it impractical to train in a normal research setting. The importance of input feature variability cannot be over-emphasized when trying to improve model training. Both global and local features contribute to model learning, and multi-scale inputs can help to improve classification accuracy for deep learning models by providing multiple views of the input at different scales.

Given these considerations, the method in this paper leverages multi-scale, and multi-label inputs, and performs resampling of an imbalanced dataset using pseudo-labels for morphological image patches to improve feature extraction during training. This approach allows for colony images containing multiple classes to be used in a patch wise manner to inform model learning by increasing the features available to the network. It is shown empirically here that this method displays improvement over standard convolutional neural networks, as well as similar methods of contrastive learning for biological datasets.

The single-scale networks used for the training configurations in this work are modified to accept multi-scale inputs. For the single scale architecture, size 128×128 image patches are extracted from the training images as input for the network. For the multi-scale configuration, the high scale input is a 224×224 image patch and the lower scale is a 112

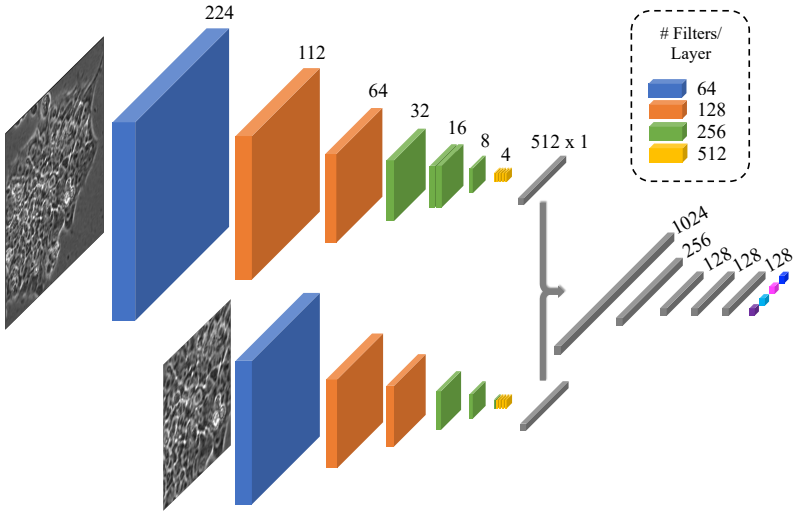

Figure 5.3: Overview of multi-scale VGG Network Architecture. Image patches of size 224 and 112 are provided as input to two separate network streams, from which feature vectors of 512 are concatenated and used to make a classification decision over the four classes. Numbers on top of features maps indicate image size at the corresponding cross section, and the legend in the top right displays the number of filter channels in each color coded layer. Batch normalization is added between every convolutional layer, as well as after the feature concatenation layer, and ReLU activation is used between all layers until the final classification.

$\times$ 112 center patch of the higher scale image. This allows for the resulting feature vectors of the two inputs to be concatenated at equal lengths before the final classification layers of the network. A diagram of the multi-scale VGG network architecture is shown in Figure 5.3, and image samples at various scales can be seen in Figure 5.4.

These input scales are determined based on the relationship between optical properties of the microscope used for data collection and the relevant optical feature scale of cellular colonies in the images. Individual cell sizes can range from 10-100 μm, and due to the combination of optical parameters of the microscope unit used for data collection in this work, the individual pixel size of the live cell images is 0.8 μm [93]. Therefore, a line of 128 pixels equates to 102.4 μm within the region of interest. The larger scale image patch,

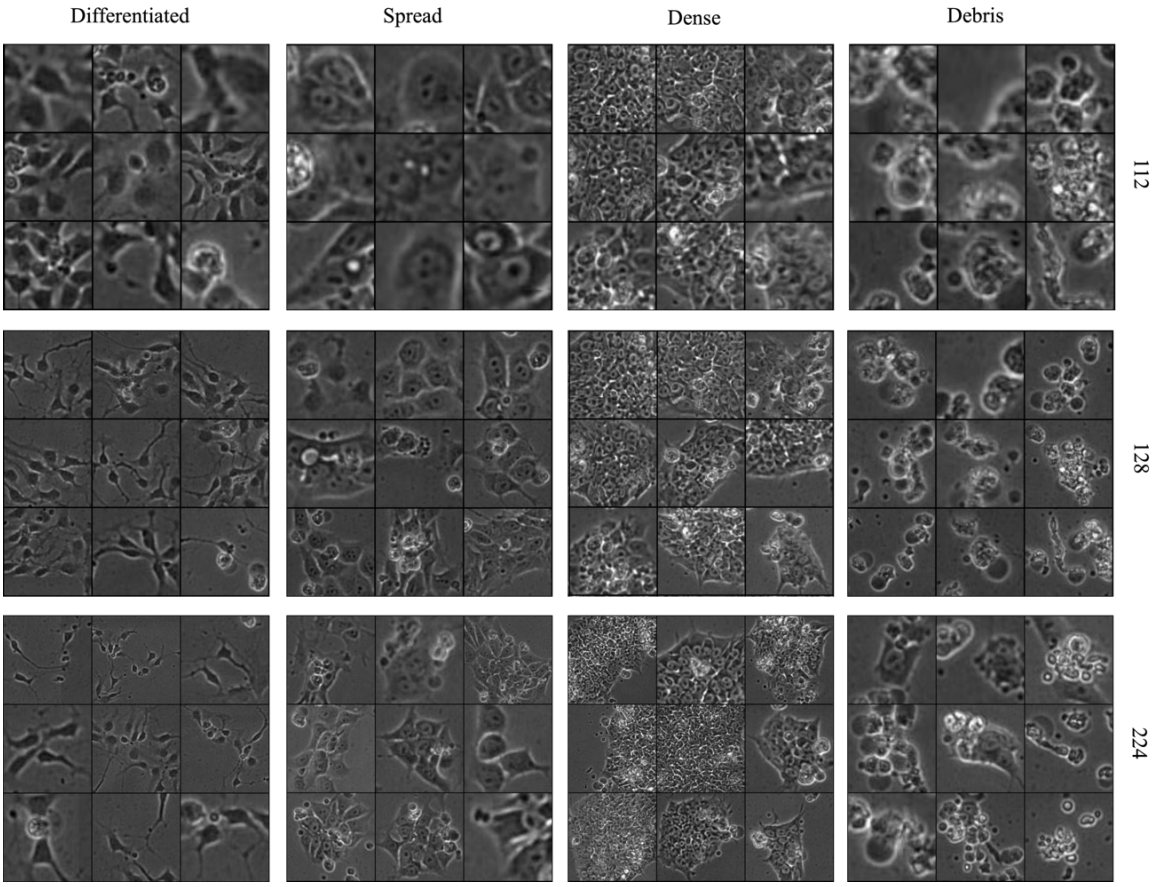

Figure 5.4: Sample image patches for every class at three scales (112, 128, and 224). Different views of image features are provided at each scale. At the lowest scale, 112 × 112, local views of fine-grain texture patterns are predominant. At the 128 × 128 scale, local texture features are present but global features, such as edges, are also observable. At the 224 × 224 scale colony shape becomes an important feature because images contain views of entire colonies.

224 × 224, encompasses a more global view of the input image (179.2 μm^2) and contains features such as colony shape, edges, and surrounding area. The smaller-scale input patch, 112 × 112 (89.6 μm^2), captures the morphological texture patterns of the cellular area within a colony. Both image patch scales contribute useful information for the network to learn, and the concatenation of learned feature vectors provides more information for the network to use in its classification decision.

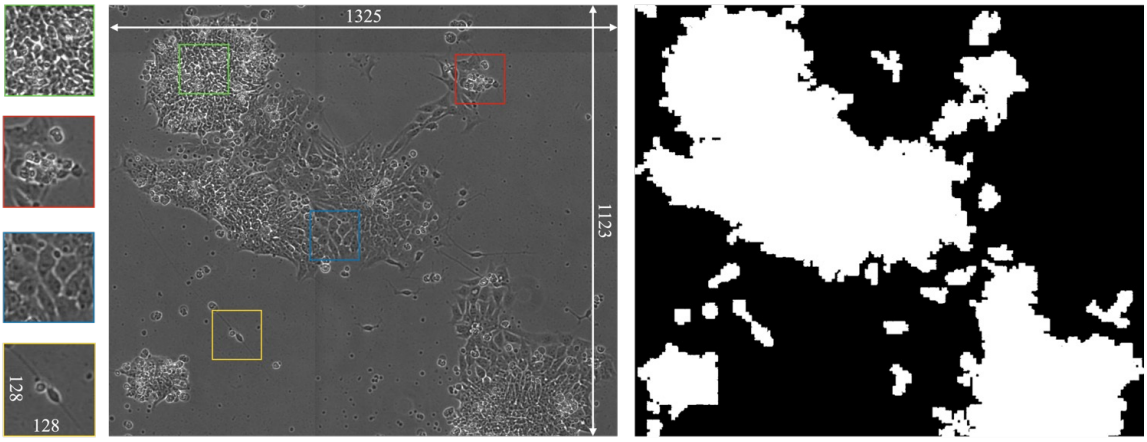

Figure 5.5: (Right) Binary map of cell colony area calculated using morphological segmentation. This map is used to reduce the presence of background area when taking image patches. (Center) Example of a large-scale (1325 × 1123) multi-label image containing areas of each of the individual four classes. (Left) 128 × 128 patches of images and their corresponding locations within the larger image (from top to bottom: Dense, Debris, Spread, Differentiated). The high background to foreground ratio makes taking random patches from within the binary map of the colony a crucial step in providing relevant information to the model.

The networks are then trained to learn features of the input patches and sort them into the four morphological classes. Some of the images contain areas of more than one image class and because they were captured without the use of stains of fluorescent biomolecules, they would not be usable in traditional deep learning training configurations which necessitate image level labels as a minimal requirement. However, using the IPB algorithm, image patches from multi-label images can be used to train the student network when given pseudo-labels by the teacher network.

Multi-label Input

Figure 5.5 provides an example of a multi-label colony image that contains areas of all four of the relevant image classes. The colonies in these images contain multiple cell

classes with contiguous boundaries. As described above, it is very difficult to provide sub-image level labels for these inputs because there exists no pixel level ground-truth in the dataset and as a result these images are useless for training traditional CNN architectures using reinforcement learning. However, the IPB algorithm provides an avenue for including image patches from multi-label images without providing semantic labels.

This is accomplished by simply providing image patches of multi-label images to the teacher network and conserving those image patch areas for each minibatch of inputs. Additionally, the MPL algorithm determines both soft pseudo-labels, as well as hard pseudo-labels provided to the student network using a prediction confidence threshold of than 0.65. In this way, any class overlap that might be captured in a random image patch can be filtered out before reaching the student network. By including multi-label images, the network is provided with more variability of class input features. Random image patches are resampled for every epoch, allowing the network to learn from a balanced dataset distribution on the fly.

Dataset Augmentations

Every random patch input is transformed using a series of random augmentations that includes horizontal and vertical flipping, 0-180 degree rotation, contrast and brightness adjustment, and gaussian blurring. Each of these augmentations is applied independently with a 0.25 probability, and they help to increase the effectiveness of the UDA module that is used to train the teacher network. Furthermore, several inherent augmentations are included in dataset pre-processing which also contribute to variability within the training dataset.

For example, patch crops provide an intrinsic randomized view of input images during each training epoch and introduce different morphological patterns and foreground to background ratios. Also, images that are too small to be cropped within the given patch size necessitate a resizing augmentation that helps account for scale invariance within the image data. These examples of innate augmentations highlight the relevance of biological variability in improving network generalization by expanding the apparent size and scope of the input dataset. Specifically those factors relating to the collection of microscopy data, which can encompass multiple scales, lighting levels, and optical parameters.

Iterative Pseudo Balancing Resampling Scheme

The pseudo label resampling scheme proposed in this work is a crucial step in balancing the input dataset for IPB learning. An overview of the pre-training, training, and testing steps are provided in previous sections. In line with the standard MPL algorithm, the pre-trained teacher provides pseudo labels to patch samples of the unlabeled training dataset. Where normally these pseudo labeled images would then be directly used to update the student network, IPB adds an intermediate step that involves balancing the pseudo-labeled image patches by weighting each class based on their relative probabilities within the dataset.

At the beginning of each epoch, the teacher network provides a pseudo label to each random image patch from the unlabeled dataset. Given a dataset with m classes, the minimum number of samples for a given class is $n_i = \min(n_1, n_2, \ldots, n_m)$ for $i = 1 \rightarrow m$. The proportion of every class in the unlabeled dataset (p_i) is determined using the pseudo-labels and samples are drawn from a multinomial distribution with replacement using the

inversely proportional weights of the classes within the dataset. The number of images taken from a given class $n_i = p_i * n_i$ such that the total number of images taken from a given class can vary given the proportion of image labels in the dataset of image patches. The resampled dataset is used to update the student network, and the student networks performance on the labeled dataset is used to update the teacher networks such that it can provide more accurate pseudo-labels to the student.

Inefficiences in network learning caused by class imbalances are due to the effect of confirmation bias in model learning [126], which is when the teacher network overfits on the most prevelent class, so that when the student is provided with random, pseudo-labeled data points, the proportion of these images coming from a given class cannot be controlled, and given the high levels of imbalance present in the dataset used for this work, the network begins to only learn features of the most prevalent class. The IPB resampling scheme proposed in this work overcomes the problem of confirmation bias due to class imbalance by using the pseudo-labels to balance the image dataset on the fly for every training iteration.

5.4 Results and Discussion

5.4.1 Dataset and Ground-truth

The input dataset for this work is composed of time-lapse, phase-contrast microscopy images of induced pluripotent stem cells. The goal of these experimental studies is to determine the developmental effects of nicotine on Huntington's Disease affected neuro-

Algorithm 1 Iterative Psuedo Balancing Algorithm

$U \leftarrow$ Unlabeled Dataset
$L \leftarrow$ Labeled Dataset
$T \leftarrow$ Testing Dataset
$NSE \leftarrow$ Number of Student Epochs
$N \leftarrow$ Number of Training Loops
procedure INITIALIZE TEACHER AND STUDENT NETWORKS
    Teacher and Student networks both use HRNet architecture
    Teacher and Student network weights are initialized with Kaiming Initialization
procedure PRE-TRAIN TEACHER NETWORK
    Balance labeled dataset using resampling based on ground-truth labels
    Update teacher network using cross-entropy loss on L with HRNet
    Repeat for N pre-training epochs
procedure TRAIN IPB NETWORK
    Collect pseudo-labels using pre-trained teacher network predictions for U
    Resample image patches in weighted proportions of class psuedo-labels obtained from the teacher network
    Update student HRNet weights in relation to cross-entropy loss between predictions of student network and
        pseudo-labels from the teacher on balanced pseudo-labeled dataset
    Collect student predictions for L image patches
    Update teacher HRNet weights in relation to student cross-entropy loss between student predictions on labeled
        image patches and ground-truth labels from L
    Repeat for all batches in U for total number of student epochs NSE
    Repeat update of teacher and student for N training loops
procedure TEST STUDENT NETWORK
    Collect student network prediction for every labeled image patch in T
    Compare student predictions on testing dataset T to target label to test classification performance

genesis. The study was designed and conducted by Dr. Barbara Davis from the Laboratory of Dr. Prue Talbot at the University of California, Riverside, Stem Cell Center. The experimental design centers around the premise that nicotine has a neuroprotective effect on cells affected by neurodegenerative diseases and involves culturing the cells under control and experimental conditions to observe their behavior over the course of 48 hours using the Nikon Biostation CT incubator/microscope unit. This microscope outputs 2908×2908

stitched images of the entire culture well at 10x optical magnification, taking one image every hour over the course of the experiment. In total, the dataset includes 15, 48 image time-lapse sequences, which are pre-processed and annotated before being used for network training.

Cropped colony ROI's from the dataset of raw microscope images are sorted into four morphological classes that correspond to the unique phenotypes that are observed during the experiment. These classes include Dense, Spread, Differentiated, and Debris, each characterized by distinct texture and shape features that translate directly to the health and developmental status of the cell colonies. More specifically, the **Dense** class corresponds to early stage pluripotent stem cells, with relatively small individual size and dense, uniform, fine grained texture; the **Spread** class indicates intermediate stage progenitor cells with larger cell body area and a more uneven texture pattern; the **Differentiated** class represents adult neurons with dark cell bodies and elongated protrusions called axons; the **Debris** class displays spherical groups of dying or dead cells with high contrast indicative of the bubble like rounding of the dead cells which leads to shiny white areas around the perimeter.

The ground-truth labels were determined by a skilled biologist along with two other individuals who were trained to distinguish between the individual classes by visual appearance and provided with exemplary image references. Final ground-truth was determined via majority vote of annotations for each colony image. Table 5.1 details the breakdown of the number of images that fall into each data class. Images range a variety of scales, with the smallest being 55×85 pixels, and the largest being 771×1298. Colony images that

Class	# Samples
Debris	3587
Dense	3934
Diff	656
Spread	10506
Multi-label	2659
Total	21342

Table 5.1: Breakdown of data samples per class for the stem cell microscopy dataset

contain areas with more than one morphological class label were sorted into a separate bin for multi-label images. These 2,659 images in the dataset were previously impractical to use for neural network training because their predicted class cannot be compared to any single ground-truth, however, patches of these images are given pseudo labels in conjunction with the rest of the dataset during IPB training. Subcategories of these images include partially differentiated colonies (1,436 images), that contain at least some portion of the differentiated class, as well as various proportions of the other downstream classes; and partially spread (1,223 images) which similarly contain some portion of spread class as well as various proportions of downstream classes. The training dataset contains both labeled and unlabeled images, where the majority of the dataset (80%) is used as unlabeled data for the semi-supervised IPB algorithm, a smaller portion (10%) is used for the supervised pre-training, fine-tuning and teacher update steps, and another small portion (10%) is used for model validation.

True Positive Rate (TPR) and F1 score are used as metrics for classification accuracy because for an imbalanced dataset, TPR delineates how misclassifications affect the performance of the network, and F1 score computes the harmonic mean of precision and

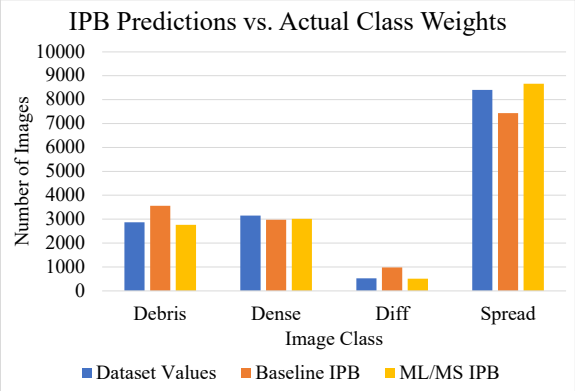

Figure 5.6: Predicted class weights from the teacher network for two IPB configurations, as well as the actual training dataset distribution. The multi-scale input patches from multi-label images (ML+MS) allows the teacher to learn a more accurate distribution of the image classes, which it can then use to provide the student with a more accurate pseudo-balanced dataset.

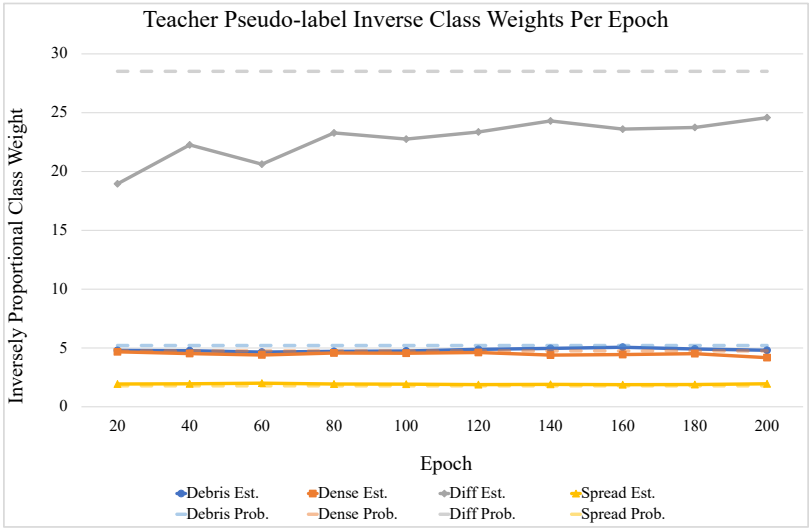

Figure 5.7: Sampling weights for each class displayed as the inverse proportion of pseudo-labels as predicted by the teacher network across the unlabeled image dataset.

recall, providing a comprehensive view of the models susceptibility to false positives and false negatives. TPR and F1 score are calculated as as shown in Equations 5.3 and 5.4. The results of the training configurations containing these images is detailed in the following sections.

$$TPR = \frac{TP}{TP + FN} \tag{5.3}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{5.4}$$

5.4.2 Pseudo-Label Resampling Scheme Accurately Predicts Class Weights for Training Dataset

The first important step in training any of the experimental IPB configurations is the resampling scheme proposed in this work. Without performing dataset balancing via pseudo label resampling as outlined in the previous section, the MPL network overfits on the most prevalent class, resulting in imbalanced classification. However, by adding this step, the student network learns normally from a balanced dataset after the teacher performs pseudo-labeling.

This is important because one of the main limitations of semi-supervised networks is the inability to learn effectively on imbalanced datasets. Furthermore, biological datasets are often highly imbalanced, so the psuedo-label resampling scheme allows for the unlabeled images to contribute equally to the multi-modal distribution learned by the neural network. This is illustrated in Figure 5.6, which displays the predicted class proportions of various networks. The proportions are determined by accumulating the teacher networks predictions across the labeled training dataset, which are then used to balance the dataset of image patches during each training epoch. Figure 5.7 shows the teachers progression in estimating the proportions of every class in the unlabeled dataset by providing each image a pseudo-label. As training progresses, the teacher begins to learn a more balanced distribution of

the dataset, as evident by the values for the differentiated class trending towards the actual class proportion.

5.4.3 Ablation Experiment: Multi-Scale and Multi-Label Inputs Provide Global Features Without the Need for Annotation

The multi-scale IPB configurations (MS HRNet, MS IPB VGG, MS IPB HRNet) which take both 224 and 112 scale inputs to create a combined feature vector, show the greatest improvement over the baseline architectures, in the Dense and Differentiated classes, especially for the HRNet model configuration. These observations may be due to the global features that are incorporated into feature learning with the multi-scale input. Specifically, the introduction of morphological features such as edges, colony shape and size, and foreground to background contrast contribute to improved feature discrimination during model learning. This includes a statistically significant increases in the TPR of the Dense class by $\sim 5\%$ (P-value: 0.0009 using student t-test with 95% confidence value) and the Differentiated class by $\sim 7\%$ (P-value 0.0448 using student t-test with 95% confidence) when compared to the baseline HRNet. For the other classes, these features are already present or prominent in the single-scale input or do not contribute additional information at a higher-scale (Figure 5.4).

The multi-label IPB configurations (ML IPB VGG, ML IPB HRNet) includes single scale input patches of unlabeled training images that contain multiple image classes within a single colony. By taking random crops of these images iteratively during training, the number of additional unique features that these images contribute is extremely large. These improvements in model learning can be attributed to the increase in available data-

Config./ Class (std.)	Debris	Dense	Diff	Spread	Average
VGG19	0.8877 (0.0174)	0.7772 (0.0308)	0.8434 (0.0396)	**0.9299 (0.0134)**	0.8595
HRNet	0.8670 (0.0087)	0.7752 (0.0291)	0.8162 (0.0569)	0.9239 (0.0076)	0.8455
MS HRNet	0.7921 (0.0491)	0.8410 (0.0116)	0.9013 (0.0595)	0.8813 (0.0111)	0.8539
IPB VGG	0.8389 (0.0615)	0.7695 (0.0590)	0.7971 (0.0621)	0.9281 (0.0277)	0.8334
MS IPB VGG	0.8843 (0.0576)	0.8489 (0.0344)	0.8369 (0.0913)	0.8949 (0.0340)	0.8662
ML IPB VGG	0.8890 (0.0356)	0.8061 (0.0467)	0.8516 (0.0605)	0.9144 (0.0190)	0.8647
ML+MS IPB VGG	0.8905 (0.0419)	0.8529 (0.0312)	0.8259 (0.0741)	0.9006 (0.0148)	0.8674
MS IPB HRNet	0.8934 (0.0512)	0.8241 (0.0295)	0.8846 (0.0826)	0.9112 (0.0174)	0.8783
ML IPB HRNet	**0.9139 (0.0364)**	0.7920 (0.0311)	0.8737 (0.0785)	0.9139 (0.0158)	0.8733
ML+MS IPB HRNet	0.8843 (0.0159)	**0.8575 (0.0218)**	**0.9085 (0.0189)**	0.8985 (0.0074)	**0.8872**

Table 5.2: True positive rate of classification for all IPB configurations, where Multi-scale is denoted as MS, and Multi-label is denoted as ML.

points for feature extraction. Another possible consideration for the multi-label images is that they represent some later stage cell colonies because of the larger colony size and presence of multiple classes. This could account for some of the improvement in the Dense class as well as classes representing the later stage phenotypes (Differentiated and Spread).

5.4.4 Combining Multi-label and Multi-Scale Inputs Maximizes Available Features for Model Learning

While the separate implementation of multi-label patches and multi-scale inputs results in significant improvement over baseline CNN and IPB implementations, the com-

Config./ Class (std.)	Debris	Dense	Diff	Spread	Average
VGG19	**0.8723 (0.0161)**	0.8342 (0.0162)	0.8460 (0.0255)	0.9053 (0.0073)	0.8644
HRNet	0.8579 (0.0111)	0.8215 (0.0154)	0.8565 (0.0174)	0.9011 (0.0055)	0.8567
MS HRNet	0.8122 (0.0206)	0.8071 (0.0066)	0.7628 (0.0541)	0.8901 (0.0072)	0.8180
IPB VGG	0.8497 (0.0245)	0.8152 (0.0142)	0.8037 (0.0678)	0.8899 (0.0143)	0.8396
MS IPB VGG	0.8529 (0.0374)	0.8339 (0.0275)	0.8190 (0.0439)	0.9044 (0.0109)	0.8525
ML IPB VGG	0.8627 (0.0228)	0.8293 (0.0189)	0.8519 (0.0300)	0.9063 (0.0060)	0.8625
ML+MS IPB VGG	0.8574 (0.0182)	**0.8513 (0.0196)**	0.8204 (0.0463)	0.9102 (0.0104)	0.8598
MS IPB HRNet	0.8608 (0.0141)	0.8412 (0.0158)	0.8536 (0.0346)	0.9125 (0.0036)	0.8670
ML IPB HRNet	0.8580 (0.0146)	0.8419 (0.0188)	0.8587 (0.0486)	0.9067 (0.0057)	0.8663
ML+MS IPB HRNet	0.8609 (0.0102)	0.8422 (0.0020)	**0.8749 (0.0278)**	**0.9132 (0.0030)**	**0.8728**

Table 5.3: F1 Score for all IPB configurations, where Multi-scale is denoted as MS, and Multi-label is denoted as ML.

bination of these modules for IPB training demonstrates the true power of these features when used in conjunction. Tables 5.2 and 5.3 contain the classification results for the IPB training configurations in relation to a standard VGG and HRNet CNN architectures in terms of TPR and F1 Score.

Overall, the ML+MS IPB HRNet improves classification accuracy as measured by TPR over the standard CNN architectures by ∼3%, with statistically significant improvements in the Dense (∼ 8%), and Differentiated (∼ 9%) Classes using a student t-test with a 95% confidence rate (p-values: 0.0001, and 0.0002, respectively). As previously stated, the introduction of global features using multi-scale inputs improves feature disentanglement

by providing context for the networks class prediction. Additionally, these improvements suggests that pseudo-balancing the multi-label and multi-scale data allows the network to learn from a more balanced distribution of class features.

In terms of F1 score, the ML+MS IPB HRnet configuration also shows the greatest improvement over the baseline CNN, by $\sim$2%, with statistically significant improvements in the Dense class via student t-test (p-value: 0.0005). These improvements may be due to the fact that HRNet has the capacity to conserve high-level features such as shape characteristics of the larger scale input as well as model texture features present at the smaller scale input patches.

These improvements can be attributed to the interplay between the multi-label and multi-scale image samples that are present in the dataset. For example, the 2,659 additional images that are introduced for the multi-label configuration tend to represent larger colonies from later-stage images. When taking single scale patch samples of these images, the surrounding area is not taken into consideration, which results in fewer available features for the network. Conversely, the global information provided by the multi-scale input may not provide as much additional information for model learning when applied to the smaller, single-class images. However, when combined, the multi-scale, multi-class input provides useful features from a large portion of the dataset that was previously unusable in a supervised network setting.

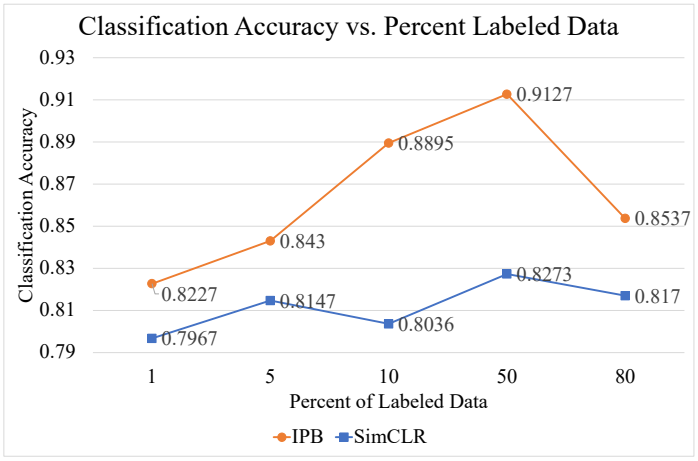

Figure 5.8: Graph of classification accuracy vs. proportion of labeled data used for training. There is a positive correlation between the amount of labeled data and the final classification accuracy of the student. However, these results still demonstrate that in a data limited setting, this method can still be effective.

Reducing Proportion of Labeled Training Data for Data Limited Settings Lowers Classification Performance

To test the ability of the model to learn on smaller proportions of labeled data for use in the teacher pre-training and MPL update steps, network configurations were trained using 1%, and 5% as well as 50% and 80% of labeled data and compared to the standard 10% configuration used in this work. The lowest data setting (1%) represents only 191 total labeled images, whereas the 5% setting contains 938 labeled images, the 10% setting contains 1,873 labeled images, the 50% configuration contains 9,341 labeled images and the 80% configuration contains 14,946 labeled images.

Figure 5.8 demonstrates a general upward trend in classification accuracy for the IPB network configuration until the 50% data mark, when for the 80% configuration, the accuracy of the model declines. This is because the student network is trained on images with pseudo-labels from the teacher network, which become more robust with increased

number of labeled data points until the amount of data used to generate confident pseudo-labels results in a lack of sufficient data points with which to train the student network effectively. However, the results using the lower data settings still highlight the potential of this method to train the neural network even in very data limited settings.

These dataset configurations are also used to train the SimCLR model, which is the most competitive related work to the proposed method. This self-supervised method has the limitations of not being able to estimate the dataset distribution, as well as requiring a large amount of data for effective feature mapping. This can be seen in Figure 5.8 where fluctuations in accuracy can be seen, with a general upward trend. Given the size of the dataset used in this work, there are not enough data points for the network to learn effectively, resulting in overall lower performance as compared to the IPB method proposed in this work, which obtains superior results to this method using only a fraction of the training data.

5.4.5 Observation of Misclassifications Highlights Importance of Multi-Scale Network

Figure 5.9 shows examples of misclassified patches for the multi-label/multi-scale configuration. Several observations can be made including that misclassifications as Debris are often due to the presence of dead or dying cells within or surrounding healthy colony areas; Spread images misclassified as Dense sometimes contain areas with more than one cell class present and vice versa for the Dense to Spread class; misclassified differentiated images are confused by the network because they can appear flat like a spread image, but

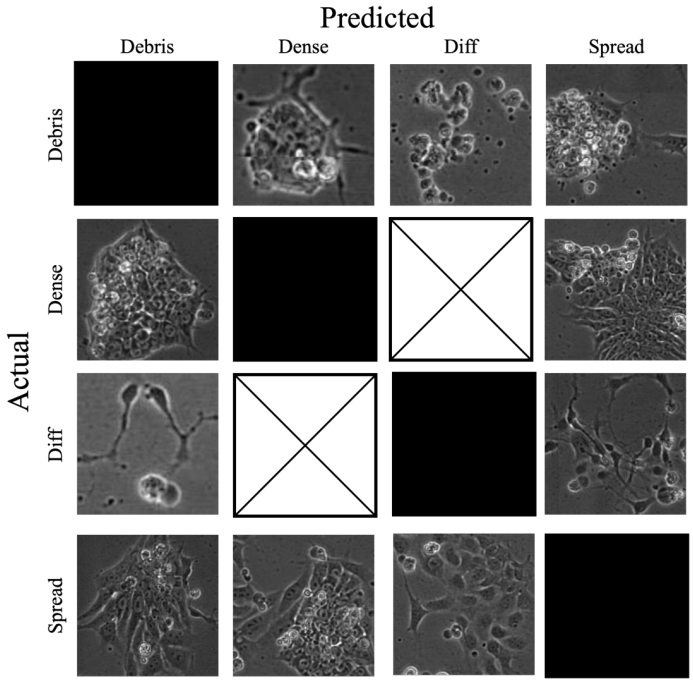

Figure 5.9: Example misclassifications for the multi-scale/multi-label configuration. Black boxes represent the omitted correct classifications, and boxes with and x through them represent instances where no misclassifications occurred. Figure 5.4 provides example images of each of the classes.

no differentiated images were improperly predicted as Dense and vice versa, mostly because these classes are the least closely associated with one another biologically, whereas the Spread class is the intermediate between the Dense and Differentiated, and is the most prevalent and most commonly confused class. Table 5.4 portrays a representative confusion matrix for these network predictions, which highlights the misclassifications caused by the presence of feature overlap between biologically adjacent classes. These observations help to emphasize the importance of incorporating a multi-scale input, which can provide a view of both the fine grained textures, and global feature patterns of the input images.

Predicted/Actual Class	Debris	Dense	Diff.	Spread
Debris	301	18	2	37
Dense	11	326	0	56
Diff.	1	0	62	2
Spread	32	39	6	973

Table 5.4: Representative confusion matrix for the IPB network. The most confused classes observed here are the Dense, Debris, and Spread classes, which can be attributed to the presence of class overlap as a result of downstream differentiation. Similarly, there are no misclassifications between the Dense and Differentiated classes because of their relatively far distance in terms of biological proximity.

5.4.6 Multi-scale Pseudo-balancing Network Out-performs State-of-the-Art

Several previous works have attempted to perform semi-supervised and self-supervised learning for natural image and biological datasets. To determine the effectiveness of these methods in relation to the proposed approach, four state-of-the-art models (MPL [122], SimCLR [114], Contrastive Clustering [116], and SimTriplet [119]) were trained on the dataset, and the results were calculated across 5 folds in Table 5.5. These results confirm that the major pitfall of these methods is that they fail to account for class imbalances in the absence of prior information. Furthermore, contrastive methods such as SimCLR work most effectively with very large amounts of data, which is counterintuitive to the tasks associated with biological image datasets that are inherently limited by the constraints of experimental research.

The MPL algorithm does not inherently account for class imbalances, and also tends to bias towards on the most prevalent class (Spread). The MPL algorithm trained with the pseudo-balancing module (MPL Balanced, Table 5.5) has a lower variance in inter-class classification accuracy in comparison to the MPL Imbalanced configuration (0.0018 vs.

Config./ Class (std.)	Debris	Dense	Diff	Spread	Avg.
SimCLR [114]	0.8559 (0.0346)	**0.8712 (0.0274)**	0.7538 (0.0592)	0.7874 (0.0213)	0.8170
CC [116]	0.5101 (0.1756)	0.3852 (0.4324)	0.2277 (0.2370)	0.3436 (0.0841)	0.3666
SimTriplet [119]	0.1726 (0.0266)	0.7160 (0.1390)	0.2246 (0.1864)	0.5423 (0.1012)	0.4138
MPL (Imbalanced) [122]	0.8158 (0.0804)	0.8423 (0.0407)	0.8818 (0.0504)	**0.9292 (0.0237)**	0.8672
MPL (Balanced)	**0.8950 (0.0106)**	0.8261 (0.0540)	0.8322 (0.0564)	0.9111 (0.0221)	0.8661
IPB (ML+MS)	0.8843 (0.0159)	0.8575 (0.0218)	**0.9085 (0.0189)**	0.8985 (0.0074)	**0.8872**

Table 5.5: True Positive Rate for related works in comparison to the multi-label, multi-scale IPB configuration, where Multi-scale is denoted as MS, and Multi-label is denoted as ML. Related methods display inferior results due to the effect of class imbalances, which causes over-fitting and high variance.

0.0024), which means that the results provide a more balanced classification performance. Furthermore, these MPL configurations show inferior performance to the IPB algorithm, which incorprates multi-scale and multi-label image patches to improve feature extraction and classification. The IPB algorithm outperforms the MPL configuration by 2% by providing the model with a balanced view of the input classes and as well as a global view of input data and incorporates image features from multi-label image patches. It also does not necessitate the large amount of data required for learning image features in a fully supervised manner because it utilizes semi-supervised pseudo-labeling to guide mapping of the input distribution.

5.5 Conclusions

The approach presented in this work, Iterative Pseudo Balancing, represents a significant improvement in semi-supervised methods for resourceful management of biological datasets. Pseudo-labels from an MPL framework are used to iteratively balance a biological image dataset on the fly, and simultaneously train a student-teacher ensemble network to accurately classify stem cell colonies. Previously unusable, multi-label images are used to increase the available data points for model learning, and multi-scale inputs are used to integrate global and local features present in the dataset. Combining these views of the input data for an HRNet results in an overall improvement of 3% over the baseline CNN in terms of TPR, which is shown to be statistically significant. The results shown in this paper highlight the importance of balanced learning in biological experimentation when employing deep neural networks.

The IPB framework proposed here overcomes the problem of confirmation bias via overfitting as a factor of dataset imbalance by providing the network with a comprehensive view of the dataset, incorporating multiple image scales and exposing hidden features from mutli-label images. The exhaustive use of all available data and domain information is crucial to improving model performance when training on biological datasets. This work allows for biological researchers to easily employ deep learning analysis to their experimental datasets without having to spend time manually labeling large datasets, and to model their data using non-invasive morphological classification without the need for molecular biomarkers. In turn, this will accelerate the pace of biological research involving predictive models. Other possible avenues of exploration for improving this work include adding

motion information for dynamics, and the collection of biomarker data for endpoint ground-truth.

Chapter 6

Conclusions and Future Work

The dissertation presented here is comprised of three main works that aim to improve classification of stem cell microscopy images using biologically informed deep learning, computer vision models. Several key issues are addressed including limited and imbalanced datasets, expensive and biased manual annotation, and feature disentanglement. Together, this body of work represents a significant advancement in the field of non-invasive, image-based experimental quantification for automating and standardizing the data analysis pipeline, making it more reliable, robust, and easier to implement in practical research settings.

In Chapter 3, Generative Adversarial Networks (GAN) are used to supplement and balance a dataset of image patches to improve feature extraction and classification of a convolutional neural network. For each morphological class generated images from a GAN are filtered through an entropy based quality control module that brings the distribution of generated image features closer to that of the real data. The generated image patches

are then added to the training dataset to balance out the classes. Images are sorted in a hierarchical manner according to relationships between image classes based on the downstream differentiation potential of stem cell colonies. It is shown that adding generated images to the dataset improves classification performance of a neural network by balancing the dataset distribution while adding more features during model learning.

In Chapter 4 triplet-net CNN's are used to accurately discern between closely related morphpological classes observed in the experimental dataset. The continuous nature of cellular differentiation and the presence of contiguous cell boundaries causes class overlap which results in feature entanglement and misclassification between visually similar classes when training neural networks using image patches. The triplet-net model implemented in this work uses a contrastive loss function to determine the distance between feature vectors of a query image and samples from other classes in the dataset, allowing the model to more accurately discern between classes with similar visual features. Furthermore, the final classification decision is guided by comparison to representative images from each class that are determined using texture features. This method accurately models the downstream differentiation process of stem cell colonies and reduces misclassifications between biologically related image classes.

In Chapter 5 Iterative Pseudo Balancing is introduced as a method for training neural networks with minimal labeled training images by employing semi-supervised learning methods. The large amount of annotated data required to effectively train deep neural networks is often difficult to obtain, and the majority of collected data is necessary for experimental evaluation. In this work, meta-learning is employed to train a semi-supervised

student-teacher network that requires only 10% labeled image data. Pseudo-labeled image patches are used to train a neural network to classify multi-label images using multi-scale input that provides global feature context. It is shown empirically that the combination of these modules improves classification accuracy over standard CNN configurations and state-of-the-art related methods. This work allows datasets to be balanced on the fly with minimal manual annotation and used to train neural networks that increase the accuracy and efficiency of experimental analysis.

These works highlight the necessity of domain-inspired deep neural networks for automated experimental analysis. Deep learning is a powerful tool for non-invasive modeling of morphological features that are undetectable by the human eye, overcoming limitations imposed by sources of human error and bias. The findings in these works reveal key insights into the underlying mechanisms of observed morphological behavior and can be used to infer relevant experimental conclusions. Nicotine is shown to increase the rate of growth of Huntington's disease expressing iPSC's, but the extent of neuroprotective and neurogenic effects requires further investigation. These approaches have implications in a variety of applications including microscopy, pathology, and biomedical image analysis, and are useful tools for researchers to accelerate the pace of scientific discovery.

Still, there exist several avenues for the expansion of this work into future projects. There is potential to improve feature extraction, modeling, and classification by designing novel network architectures and addressing limitations imposed by the dataset. For example, molecular biomarker validation, unavailable for the phase-contrast dataset examined here, could be used to determine pixel level features of cellular morphology for semantic

segmentation. Furthermore, multi-modal information such as protein expression, RNA and DNA sequencing would improve understanding of underlying biological mechanisms, allowing for morphological behavior to predict molecular level changes. Moreover, while temporal relationships between classes were exploited to inform model design, dynamic colony behavior was not directly modeled between consecutive image frames. For this, cell colony tracking could be performed incorporate motion information into the training of a recurrent neural network. Finally, the methods presented in this dissertation can be extended into other imaging and data modalities including histology, medical imaging, drug design, and 3D imaging. The generalizability of these networks implies their ability to accurately quantify image data from various sources and improve functional workflow via automated classification. These improvements represent several frontiers in data science and machine learning that have the potential to transform the landscape of precision, regenerative medicine and experimental research.

Bibliography

[1] Adam Witmer and Bir Bhanu. Generative adversarial networks for morphological–temporal classification of stem cell images. *Sensors*, 22(1):206, 2021.

[2] Adam Witmer, Rajkumar Theagarajan, and Bir Bhanu. Triplet-net classification of contiguous stem cell microscopy images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

[3] Adam Witmer and Bir Bhanu. Iterative pseudo balancing for stem cell microscopy image classification. *Scientific Reports*, 14(1):4489, 2024.

[4] Atena Zahedi, Vincent On, Sabrina C Lin, Brett C Bays, Esther Omaiye, Bir Bhanu, and Prue Talbot. Evaluating cell processes, quality, and biomarkers in pluripotent stem cells using video bioinformatics. *PLoS One*, 11(2):e0148642, 2016.

[5] Adam Witmer and Bir Bhanu. Multi-label classification of stem cell microscopy images using deep learning. In *24th International Conference on Pattern Recognition (ICPR)*, pages 1408–1413. IEEE, 2018.

[6] Adam Witmer and Bir Bhanu. Hescnet: A synthetically pre-trained convolutional neural network for human embryonic stem cell colony classification. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2441–2445. IEEE, 2018.

[7] James A Thomson, Joseph Itskovitz-Eldor, Sander S Shapiro, Michelle A Waknitz, Jennifer J Swiergiel, Vivienne S Marshall, and Jeffrey M Jones. Embryonic stem cell lines derived from human blastocysts. *Science*, 282(5391):1145–1147, 1998.

[8] Hans Clevers. Modeling development and disease with organoids. *Cell*, 165(7):1586–1597, 2016.

[9] Fred H Gage. Mammalian neural stem cells. *Science*, 287(5457):1433–1438, 2000.

[10] Mark F Pittenger, Alastair M Mackay, Stephen C Beck, Rama K Jaiswal, Robin Douglas, Joseph D Mosca, Mark A Moorman, Donald W Simonetti, Stewart Craig, and Daniel R Marshak. Multilineage potential of adult human mesenchymal stem cells. *Science*, 284(5411):143–147, 1999.

[11] Jihoon Kim, Bon-Kyoung Koo, and Juergen A Knoblich. Human organoids: model systems for human biology and medicine. *Nature Reviews Molecular Cell Biology*, 21(10):571–584, 2020.

[12] Kazutoshi Takahashi, Koji Tanabe, Mari Ohnuki, Megumi Narita, Tomoko Ichisaka, Kiichiro Tomoda, and Shinya Yamanaka. Induction of pluripotent stem cells from adult human fibroblasts by defined factors. *Cell*, 131(5):861–872, 2007.

[13] Patrick S Stumpf, Rosanna CG Smith, Michael Lenz, Andreas Schuppert, Franz-Josef Müller, Ann Babtie, Thalia E Chan, Michael PH Stumpf, Colin P Please, Sam D Howison, et al. Stem cell differentiation as a non-markov stochastic process. *Cell Systems*, 5(3):268–282, 2017.

[14] Shinya Yamanaka. Pluripotent stem cell-based cell therapy—promise and challenges. *Cell Stem Cell*, 27(4):523–531, 2020.

[15] Prue Talbot, N Zur Nieden, Sabrina Lin, Ivann Martinez, Ben Guan, and Bir Bhanu. Use of video bioinformatics tools in stem cell toxicology. *Handbook of Nanotoxicology, Nanomedicine and Stem Cell Use in Toxicology*, pages 379–402, 2014.

[16] Bir Bhanu and Prue Talbot. Video bioinformatics: From live imaging to knowledge. *1st edn. Springer International Publishing. XLIII, Cham*, 381, 2015.

[17] https://www.nikon.com/products/microscope-solutions/special/ct/.

[18] Michael D Abràmoff, Paulo J Magalhães, and Sunanda J Ram. Image processing with imagej. *Biophotonics International*, 11(7):36–42, 2004.

[19] https://www.nikon.com/products/microscope-solutions/lineup/integrated/cl-quant/.

[20] Benjamin X Guan, Bir Bhanu, Prue Talbot, and Sabrina Lin. Bio-driven cell region detection in human embryonic stem cell assay. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(3):604–611, 2014.

[21] Tânia Perestrelo, Weitong Chen, Marcelo Correia, Christopher Le, Sandro Pereira, Ana S Rodrigues, Maria I Sousa, João Ramalho-Santos, and Denis Wirtz. Pluri-iq: Quantification of embryonic stem cell pluripotency through an image-based analysis software. *Stem Cell Reports*, 11(2):607, 2018.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[25] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology*, 12(11):e1005177, 2016.

[26] Eric M Christiansen, Samuel J Yang, D Michael Ando, Ashkan Javaherian, Gaia Skibinski, Scott Lipnick, Elliot Mount, Alison O'Neil, Kevan Shah, Alicia K Lee, et al. In silico labeling: Predicting fluorescent labels in unlabeled images. *Cell*, 173(3):792–803, 2018.

[27] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(3):283–292, 2018.

[28] Claire Lifan Chen, Ata Mahjoubfar, Li-Chia Tai, Ian K Blaby, Allen Huang, Kayvan Reza Niazi, and Bahram Jalali. Deep learning in label-free cell classification. *Scientific Reports*, 6:21471, 2016.

[29] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE Transactions on Medical Imaging*, 35(5):1240–1251, 2016.

[30] Padmaja Jonnalagedda, Daniel Schmolze, and Bir Bhanu. [regular paper] mvpnets: Multi-viewing path deep learning neural networks for magnification invariant diagnosis in breast cancer. In *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 189–194. IEEE, 2018.

[31] Benjamin Guan, Bir Bhanu, Rajkumar Theagarajan, Hengyue Liu, Prue Talbot, and Nikki Weng. Human embryonic stem cell classification: random network with autoencoded feature extractor. *Journal of Biomedical Optics*, 26(5):052913, 2021.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[34] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[35] Yann Lecun, Corinna Cortes, and Chris Burges. The mnist database. http://yann.lecun.com/exdb/mnist/.

[36] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:101552, 2019.

[37] Dong Nie, Roger Trullo, Jun Lian, Li Wang, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering*, 2018.

[38] Michael Majurski, Petru Manescu, Sarala Padi, Nicholas Schaub, Nathan Hotaling, Carl Simon Jr, and Peter Bajcsy. Cell image segmentation using generative adversarial networks, transfer learning, and augmentations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[39] Dali Wang, Zheng Lu, Yichi Xu, Zi Wang, Anthony Santella, and Zhirong Bao. Cellular structure image classification with small targeted training samples. *IEEE Access*, 7:148967–148974, 2019.

[40] Yair Rivenson, Tairan Liu, Zhensong Wei, Yibo Zhang, Kevin de Haan, and Aydogan Ozcan. Phasestain: the digital staining of label-free quantitative phase microscopy images using deep learning. *Light: Science & Applications*, 8(1):1–11, 2019.

[41] S. Lee, S. Han, P. Salama, K. W. Dunn, and E. J. Delp. Three dimensional blind image deconvolution for fluorescence microscopy using generative adversarial networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 538–542, 2019.

[42] Oleksandr Bailo, DongShik Ham, and Young Min Shin. Red blood cell image generation for data augmentation using conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[43] Peter Goldsborough, Nick Pawlowski, Juan C Caicedo, Shantanu Singh, and Anne Carpenter. Cytogan: generative modeling of cell images. *bioRxiv*, page 227645, 2017.

[44] Narita Pandhe, Balazs Rada, and Shannon Quinn. Generative spatiotemporal modeling of neutrophil behavior. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 969–972. IEEE, 2018.

[45] Rajkumar Theagarajan and Bir Bhanu. Deephesc 2.0: Deep generative multi adversarial networks for improving the classification of hesc. *PloS One*, 14(3), 2019.

[46] Anton Osokin, Anatole Chessel, Rafael E Carazo Salas, and Federico Vaggi. Gans for biological image synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2233–2242, 2017.

[47] Kavitha Shaga Devan, Paul Walther, Jens von Einem, Timo Ropinski, Hans A Kestler, and Clarissa Read. Improved automatic detection of herpesvirus secondary envelopment stages in electron microscopy by augmenting training data with synthetic labelled images generated by a generative adversarial network. *Cellular Microbiology*, 23(2):e13280, 2021.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[49] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019.

[50] Panagiotis Dimitrakopoulos, Giorgos Sfikas, and Christophoros Nikou. Ising-gan: annotated data augmentation with a spatially constrained generative adversarial network. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1600–1603. IEEE, 2020.

[51] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[52] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[53] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR, 2019.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[55] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.

[56] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[57] Francis O Walker. Huntington's disease. *The Lancet*, 369(9557):218–228, 2007.

[58] Joseph B Martin and James F Gusella. Huntingtons disease. *New England Journal of Medicine*, 315(20):1267–1276, 1986.

[59] Maryka Quik. Smoking, nicotine and parkinson's disease. *Trends in Neurosciences*, 27(9):561–568, 2004.

[60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[61] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[62] Ming Yang. Imbalanced dataset sampler. https://github.com/ufoym/imbalanced-dataset-sampler.

[63] Yuehua Jiang, Balkrishna N Jahagirdar, R Lee Reinhardt, Robert E Schwartz, C Dirk Keene, Xilma R Ortiz-Gonzalez, Morayma Reyes, Todd Lenvik, Troy Lund, Mark Blackstad, et al. Pluripotency of mesenchymal stem cells derived from adult marrow. *Nature*, 418(6893):41–49, 2002.

[64] Gary K Owens, Meena S Kumar, and Brian R Wamhoff. Molecular regulation of vascular smooth muscle cell differentiation in development and disease. *Physiological Reviews*, 84(3):767–801, 2004.

[65] Charles E Murry and Gordon Keller. Differentiation of embryonic stem cells to clinically relevant populations: lessons from embryonic development. *Cell*, 132(4):661–680, 2008.

[66] Brent A Reynolds and Samuel Weiss. Generation of neurons and astrocytes from isolated cells of the adult mammalian central nervous system. *Science*, 255(5052):1707–1710, 1992.

[67] Chunmei Zhao, Wei Deng, and Fred H Gage. Mechanisms and functional implications of adult neurogenesis. *Cell*, 132(4):645–660, 2008.

[68] Kirsty L Spalding, Olaf Bergmann, Kanar Alkass, Samuel Bernard, Mehran Salehpour, Hagen B Huttner, Emil Boström, Isabelle Westerlund, Céline Vial, Bruce A Buchholz, et al. Dynamics of hippocampal neurogenesis in adult humans. *Cell*, 153(6):1219–1227, 2013.

[69] David S Olton, James T Becker, and Gail E Handelmann. Hippocampus, space, and memory. *Behavioral and Brain sciences*, 2(3):313–322, 1979.

[70] SK Singhrao, JW Neal, BP Morgan, and P Gasque. Increased complement biosynthesis by microglia and complement activation on neurons in huntington's disease. *Experimental Neurology*, 159(2):362–376, 1999.

[71] Faye Begeti, Laetitia C Schwab, Sarah L Mason, and Roger A Barker. Hippocampal dysfunction defines disease onset in huntington's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 87(9):975–981, 2016.

[72] Shagun R Mehta, Colton M Tom, Yizhou Wang, Catherine Bresee, David Rushton, Pranav P Mathkar, Jie Tang, and Virginia B Mattis. Human huntington's disease ipsc-derived cortical neurons display altered transcriptomics, morphology, and maturation. *Cell Reports*, 25(4):1081–1096, 2018.

[73] Mohammad Tariq, Haseeb Ahmad Khan, Ibrahim Elfaki, Saleh Al Deeb, and Khalaf Al Moutaery. Neuroprotective effect of nicotine against 3-nitropropionic acid (3-np)-induced experimental huntington's disease in rats. *Brain Research Bulletin*, 67(1-2):161–168, 2005.

[74] Isaac Túnez, Pedro Montilla, M Carmen Muñoz, and René Drucker-Colín. Effect of nicotine on 3-nitropropionic acid-induced oxidative stress in synaptosomes. *European Journal of Pharmacology*, 504(3):169–175, 2004.

[75] Ailsa L McGregor, Jo Dysart, Malcolm D Tingle, Bruce R Russell, Rob R Kydd, and Gregory Finucane. Varenicline improves motor and cognitive symptoms in early huntington's disease. *Neuropsychiatric Disease and Treatment*, 12:2381, 2016.

[76] Justin W Kenney and Thomas J Gould. Modulation of hippocampus-dependent learning and synaptic plasticity by nicotine. *Molecular Neurobiology*, 38(1):101–121, 2008.

[77] Dana Zeid, Munir G Kutlu, and Thomas J Gould. Differential effects of nicotine exposure on the hippocampus across lifespan. *Current Neuropharmacology*, 16(4):388–402, 2018.

[78] Ariel Waisman, Alejandro La Greca, Alan M Möbbs, María Agustina Scarafía, Natalia L Santín Velazque, Gabriel Neiman, Lucía N Moro, Carlos Luzzani, Gustavo E Sevlever, Alejandra S Guberman, et al. Deep learning neural networks highly predict very early onset of pluripotent stem cell differentiation. *Stem Cell Reports*, 12(4):845–859, 2019.

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[80] Felix Buggenthin, Florian Buettner, Philipp S Hoppe, Max Endele, Manuel Kroiss, Michael Strasser, Michael Schwarzfischer, Dirk Loeffler, Konstantinos D Kokkaliaris, Oliver Hilsenbeck, et al. Prospective identification of hematopoietic lineage choice by deep learning. *Nature Methods*, 14(4):403–406, 2017.

[81] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

[82] Kathryn A Rosowski, Aaron F Mertz, Samuel Norcross, Eric R Dufresne, and Valerie Horsley. Edges of human embryonic stem cell colonies display distinct mechanical properties and differentiation potential. *Scientific Reports*, 5:14218, 2015.

[83] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.

[84] Philipp J Schubert, Sven Dorkenwald, Michał Januszewski, Viren Jain, and Joergen Kornfeld. Learning cellular morphology with neural networks. *Nature Communications*, 10(1):1–12, 2019.

[85] Krati Gupta, Daksh Thapar, Arnav Bhavsar, and Anil K Sao. Deep metric learning for identification of mitotic patterns of hep-2 cell images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[86] Wenting Chen, Shuang Yu, Junde Wu, Kai Ma, Cheng Bian, Chunyan Chu, Linlin Shen, and Yefeng Zheng. Tr-gan: topology ranking gan with triplet loss for retinal artery/vein classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 616–625. Springer, 2020.

[87] Weixian Lei, Rong Zhang, Yang Yang, Ruixuan Wang, and Wei-Shi Zheng. Class-center involved triplet loss for skin disease classification on imbalanced data. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2020.

[88] Phawis Thammasorn, Wanpracha Art Chaovalitwongse, Landon Wootton, Eric Ford, and Matthew Nyflot. Deep convolutional triplet network for quantitative medical image analysis with comparative case study of gamma image classification. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1119–1122. IEEE, 2017.

[89] Mhd Hasan Sarhan, Shadi Albarqouni, Mehmet Yigitsoy, Nassir Navab, and Abouzar Eslami. Multi-scale microaneurysms segmentation using embedding triplet loss. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–182. Springer, 2019.

[90] Zhiwen Huang, Quan Zhou, Xingxing Zhu, and Xuming Zhang. Batch similarity based triplet loss assembled into light-weighted convolutional neural networks for medical image classification. *Sensors*, 21(3):764, 2021.

[91] Umut Uludag, Arun Ross, and Anil Jain. Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, 37(7):1533–1542, 2004.

[92] Laurent SV Thomas and Jochen Gehrig. Multi-template matching: a versatile tool for object-localization in microscopy images. *BMC bioinformatics*, 21(1):1–8, 2020.

[93] LE Wadkin, LF Elliot, I Neganova, NG Parker, V Chichagova, G Swan, A Laude, M Lako, and A Shukurov. Dynamics of single human embryonic stem cells and their pairs: a quantitative analysis. *Scientific reports*, 7(1):1–12, 2017.

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[95] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[96] R. Theagarajan and B. Bhanu. An automated system for generating tactical performance statistics for individual soccer players from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2):632–646, 2021.

[97] Adam Witmer, Rajkumar Theagarajan, and Bir Bhanu. Supplemental material for manuscript: Triplet-net classification of contiguous stem cell microscopy images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

[98] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[99] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):610–621, 1973.

[100] Alceu Ferraz Costa, Gabriel Humpire-Mamani, and Agma Juci Machado Traina. An efficient algorithm for fractal analysis of textures. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 39–46. IEEE, 2012.

[101] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2):169–190, 2017.

[102] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[103] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11):e21, 2019.

[104] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[105] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016.

[106] Jonas Cerneckis and Yanhong Shi. Context matters: hpsc-derived microglia thrive in a humanized brain environment in vivo. *Cell Stem Cell*, 30(7):909–910, 2023.

[107] Alexander H Laperle, V Alexandra Moser, Pablo Avalos, Bin Lu, Amanda Wu, Aaron Fulton, Stephany Ramirez, Veronica J Garcia, Shaughn Bell, Ritchie Ho, et al. Human ipsc-derived neural progenitor cells secreting gdnf provide protection in rodent models of als and retinal degeneration. *Stem Cell Reports*, 2023.

[108] Monia Barnat, Mariacristina Capizzi, Esther Aparicio, Susana Boluda, Doris Wennagel, Radhia Kacher, Rayane Kassem, Sophie Lenoir, Fabienne Agasse, Barbara Y Braz, et al. Huntington's disease alters human neurodevelopment. *Science*, 369(6505):787–793, 2020.

[109] Maria Jimenez-Sanchez, Floriana Licitra, Benjamin R Underwood, and David C Rubinsztein. Huntington's disease: mechanisms of pathogenesis and therapeutic strategies. *Cold Spring Harbor Perspectives in Medicine*, 7(7):a024240, 2017.

[110] Pei Teng Lum, Mahendran Sekar, Siew Hua Gan, Srinivasa Reddy Bonam, and Mohd Farooq Shaikh. Protective effect of natural products against huntington's disease: an overview of scientific evidence and understanding their mechanism of action. *ACS Chemical Neuroscience*, 12(3):391–418, 2021.

[111] Atena Zahedi, Vincent On, Rattapol Phandthong, Angela Chaili, Guadalupe Remark, Bir Bhanu, and Prue Talbot. Deep analysis of mitochondria and cell health using machine learning. *Scientific reports*, 8(1):1–15, 2018.

[112] Benjamin E Reubinoff, Martin F Pera, Chui-Yee Fong, Alan Trounson, and Ariff Bongso. Embryonic stem cell lines from human blastocysts: somatic differentiation in vitro. *Nature Biotechnology*, 18(4):399–404, 2000.

[113] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[114] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020.

[115] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in Neural Information Processing Systems*, 33:8765–8775, 2020.

[116] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8547–8555, 2021.

[117] Michael Murphy, Stefanie Jegelka, and Ernest Fraenkel. Self-supervised learning of cell type specificity from immunohistochemical images. *Bioinformatics*, 38(Supplement_1):i395–i403, 2022.

[118] Biraja Ghoshal, Feria Hikmet, Charles Pineau, Allan Tucker, and Cecilia Lindskog. Deephistoclass: a novel strategy for confident classification of immunohistochemistry images using deep learning. *Molecular & Cellular Proteomics*, 20, 2021.

[119] Quan Liu, Peter C Louis, Yuzhe Lu, Aadarsh Jha, Mengyang Zhao, Ruining Deng, Tianyuan Yao, Joseph T Roland, Haichun Yang, Shilin Zhao, et al. Simtriplet: Simple triplet representation learning with a single gpu. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 102–112. Springer, 2021.

[120] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[121] Bárbara C Benato, Alexandru C Telea, and Alexandre X Falcão. Deep feature annotation by iterative meta-pseudo-labeling on 2d projections. *Pattern Recognition*, 141:109649, 2023.

[122] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021.

[123] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[124] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[125] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3349–3364, 2020.

[126] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.