Using Generative Adversarial Networks to create multi-channel images of cells undergoing macropinocytosis

D. Catchpole, N. Shkeir and A. Smith.

Mathematics of Real-World Systems CDT, Mathematics Institute, University of Warwick

June 2020

Abstract

Subcellular protein dynamics captured by high resolution fluorescent microscopy of molecular labels present complex training data for deep learning. The process of macropinocytosis involves multiple proteins that regulate the actin cytoskeleton, however not all of them can be imaged concurrently, thus posing a challenge when attempts are made to investigate the underlying causes of the process. This paper studies the relationship of multi-channel protein distributions within cells through the application of a conditional generative adversarial network, where a target protein distribution is learned from a reference protein. The model aims to learn a one-to-one mapping between protein markers within a cell, from which a realistic 3D multi-channel image of a virtual cell can be built. Multiple trained GAN models with the same input reference protein can be used to help map the spatio-temporal order of actin recruitment within a cell. As actin and its regulators are closely associated with the cell membrane, a method of processing protein distributions on or close to the cell membrane is also presented. The method is evaluated on 2D slice, 2D surface and 3D volume data, the latter is then compared with a pseudo 3D approach that combines a set of layered 2D slices to build a 3D volume. As a result, the pseudo 3D generators showcase a significant loss of critical detail when compared to the computationally less efficient full 3D model, especially in the z-direction where there is no convolution in the pseudo 3D model.

1 Introduction

Identifying the interactions between proteins on an intra-cellular level is key to understanding how various cellular processes work. The process used to obtain images of multiple proteins within a cell is complex with several key issues, the main one being that it is often infeasible or even experimentally impossible to image multiple proteins within a cell at the same time. This limitation often allows us to only obtain either a single channel image consisting of a single protein or two channel images in the case where the two proteins can be imaged together.

This has led to a lack of multi-channel data to analyse and model various cell processes, for example in this paper we will discuss the proteins involved in macropinocytosis within *Dictyostelium* cells. There are often correlations between the proteins within a cell, especially where the two proteins are both involved in the same cell process, and this information can be used to learn a mapping between proteins so that models of multi-channel images can be generated from dual colour data using a common reference label.

In this paper we will approach this problem by using generative adversarial networks (GANs) in multiple different contexts. We first start with training a GAN to identify and learn the mapping between labels in simple yeast cells in two dimensions. We then move onto a more complicated proof of concept, learning the mapping between more complex proteins in a completely generated mock dataset. In moving closer towards three dimensions, cell surface data extracted from images of real *Dictyostelium* cells is used to learn a mapping between the spherical projection of two protein markers, PIP3 and LifeAct, a label for filamentous actin.

Moving to full three dimensions, a conditional, fully convolutional 3D GAN is tested on 3D cell data with red and green marked proteins. The model learns the distribution of the green fluorescent marker protein in 3D based on the distribution of the red protein. This 3D approach will then be compared with results from a pseudo-3D volume constructed with 2D outputs from an existing neural network (pix2pix) model to evaluate if the latter is a reasonable and more efficient substitute for the 3D GAN.

One of the main applications for this work is in evolu-

tionary modelling, where the predictions could be used to generate an evolutionary model of a macropinosome, potentially with necessary simplifications to its geometry. In this application, the mapping which has been learnt during the training of a GAN would be useful in giving an improved and clearer mechanistic understanding of the cellular processes. The mapping that we learn using the methods defined in this paper is solely that of a mapping between the fluorescence distributions of the proteins and their markers obtained through imaging, and it is important to note that this information would need to be complemented with other features and measurements, such as cell surface movements and the curvature of the cell, for further understanding of the mechanisms involved.

2 Related work

Rapid advancements in microscopic cell imaging using fluorescent molecular labelling of intracellular proteins have allowed for highly complex and dynamic cell structures to be captured in high resolution images. The focus of this paper is the protein configuration of *Dictyostelium* cells undergoing macropinocytosis, or 'cell drinking'. The process involves the deformation of the cell membrane via fast, actin driven protrusions, forming cup-like structures which capture and engulf macroscopic fluid vesicles into the cytoskeleton. Macropinocytosis is performed by many cells in the human body, such as those involved in immune response [1] and cancer cells, and can be exploited for drug delivery, virus uptake inhibition [2] or cancer treatment [3].

The study uses data from *Dictyostelium* cells, which are very similar to immune cells as far as regulation of the actin system is concerned but allow for easier fluorescent protein marking. Macropinocytosis is non-selective but is thought to be a result of multiple proteins working together, and it is the relationship between them that is of interest. Fluorescent markers allow for the imaging of the protein configurations in 3D.

Due to the scientific importance of analysing and understanding cellular processes such as macropinocytosis, there is a great deal of research into image processing techniques on high resolution images. One example of this is in quantifying cell shapes and protein dynamics on the leading edge of a cell [4]. Cell motility is related to the process of macropinocytosis, as it involves actin driven deformations of the cell membrane. The paper measures the velocity of the leading edge of a cell through mappings between various time steps, where several contours have been fit to the cell using the image intensity. Changes in protein concentrations can be analysed using more mathematically based techniques for example on the image intensity at a given angle.

Similarly, various proteins present in cell spreading have been identified through other computational techniques [5]. This research focused on identifying the proteins involved in changing the shape of the cells by creating quantitative tools to quickly classify large amounts of data. This used both corner detection and Fourier analysis to quickly identify the difference between cell shapes and find the proteins involved in the processes driving these changes. This also includes actin, which is also present when deforming the cell structure in our case of macropinocytosis.

Deep unsupervised learning techniques and in particular convolutional neural networks (CNNs) have allowed rapid advancements in the representation ability of large sets of data using generative models such as autoencoders [6], general adversarial networks (GANs) [7] and variational autoencoders (VAEs) [8]. These models have been widely implemented in biological imaging for the purpose of data augmentation, as a consequence of a limited amount of data being available. Johnson et al.[9] proposed a method of using an autoencoder for modelling cell shapes and reducing the dimensionality of high-dimensional experimental data. Semi-supervised VAEs have recently been proposed as an alternative to unsupervised learning for single cell data with results supporting more biologically meaningful representations [10].

GANs have recently been by far the most popular generative model for biological image data augmentation [11, 12, 13]. Osokin et al.[11] proposed an adapted model of a DCGAN [14] for generating 2D synthetic yeast cell data using unseen images as well as evaluating the results using classifier two sample tests (C2ST)[15]. An image-to-image translation conditional GAN (pix2pix) [16] was recently used to generate 2D data from a binary mask input which was shown to produce visually realistic data [12].

There has been a plethora of research for GAN applications and theory, however, there has been much less research on evaluation of results that are generated by GANs. A lot of effort has gone into assessing a variety of metrics [17, 18, 19], however, a consensus has not been reached on the most appropriate. For general GAN applications, the inception score [20] and Fréchet inception distance [21] have been the most common evaluation methods to date. For augmentation of biological data, comparison of Haralick's texture features [22] have been widely adopted as evaluation measures for generated data quality [23, 12, 24].

An alternative approach for data augmentation that does not use neural networks is the use of Perlin noise, [25], specifically designed to aid in the creation of 'natural appearing' textures, attempting to capture the various complexities which arise in nature.

Perlin noise was originally developed for CGI in films and had a quick uptake in various areas of video game development, for example texturing and realistic terrain generation. More recently however it has started being used for more scientific applications, for example generating realistic looking clouds in outdoor settings [26] or in a more biological context, generating simulated breast tissue using fractal Perlin noise [27]. These are just a selection of a wide range of applications for Perlin noise and as such it has potential in this project, where it can be used to generate natural looking cell masks in three dimensions.

Calls for generative models for 3D representation in areas such as vision, robotics and medicine have driven successes by utilising a GAN approach. Models such as those for cell mask generation [28], 3D reconstruction from 2D and 2.5D image data [29, 30] and generation of 3D point clouds [31] have been developed, showing the effective results these GAN models can have with reasonable success. As an extension of the pix2pix approach, conditional GAN models for pseudo-3D have been developed [12] that favour efficiency over variability over all three dimensions.

3 Method

3.1 Data and pre-processing

There are two fluorescence microscopy datasets that we used in this report, yeast cell data that was obtained from Osokin et al. [11] and five dimensional hyperstacks of *Dictyostelium* cells obtained in collaboration with our industrial partner 3i. In the yeast cell dataset, each image contains a single cell that is composed of a red channel which is the reference protein marker and a green channel which is the protein marker of interest. The reference marker in the yeast cell dataset corresponds to a protein called Bgs4, which localises in areas of cell growth, and the green marker corresponds to a protein called Act1, which controls cellular polarity. The resolution of the yeast cell images is 96×190 (which did not have to be resized as it is a factor of 2). The only pre-processing carried out on this dataset was splitting each 2-channel image into a two single channel images which were then concatenated into a single 96×380 image pair as that is the format required for pix2pix.

For *Dictyostelium* cell volume data, we had access to five hyperstacks with each hyperstack containing data about a single cell through time. The hyperstacks consisted of 2-channel (red and green) three dimensional volumes at up to 180 frames. The red marker corresponds to the protein PIP3, a signalling molecule in the cell membrane that defines the origin of macropinocytotic cups, and the green marker corresponds to Actin. The dimensions of each hyperstack were not consistent, therefore for 2D GANs, each slice of every frame of each hyperstack was rescaled to 256×256 using resampling with area interpolation.

For the data in 3D, the data was processed in a similar way to 2D data, so the red and green channels will occupy two separate halves of the volume. Due to the inconsistent resolutions of the available hyperstacks, the data was reshaped to 119x324x248, an average of the dimensions which helps preserve the length-scale of the biological structures such as membrane protrusions and engulfed vesicles, then two channels were concatenated in the y-direction. The 119x648x248 volume has a single channel to cut down on size. This dimensionality is sufficient as the code resizes the volume data before feeding it into a GAN.

3.2 Conditional GAN

A generative adversarial network (GAN) is a generative model where two convolutional neural networks partake in a zero sum game. A generator, G, generates a fake image which is intended to come from the training data distribution and a discriminator, D, attempts to distinguish between the real and fake images [7]. In a traditional GAN, the model learns a mapping from a prior random noise distribution $p_z(\mathbf{z})$ to an output image $\mathbf{y}$, $G : \mathbf{z} \mapsto \mathbf{y}$. Whereas, in a conditional GAN (cGAN), the model learns a mapping from prior noise $p_z(\mathbf{z})$ and extra information, $\mathbf{x}$, which could be class label or image to $\mathbf{y}$, $G : \{\mathbf{x}, \mathbf{z}\} \mapsto \mathbf{y}$.

The objective function for a cGAN can be represented as a minimax game between G and D

$$\min_G \max_D (\mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))]). \tag{1}$$

The software used in this project was pix2pix [16] that implements a cGAN with an observed image $\mathbf{x}$ as input which in our case is the (red) reference protein marker image.

3.2.1 2D and 3D pix2pix architecture

Isola et al. [16] found that random noise, $\mathbf{z}$, provided to the generator is learnt to be ignored and is therefore not used in pix2pix. However with no input noise, the mapping becomes deterministic. To overcome this, noise is substituted by dropout during both training and testing phases. Dropout involves randomly ignoring nodes in the layers of the generator to add minor stochasticity to the generated samples which also reduces the likelihood of overfitting.

The cGAN in pix2pix consists of a U-net generator and a PatchGAN discriminator. The U-net generator has an

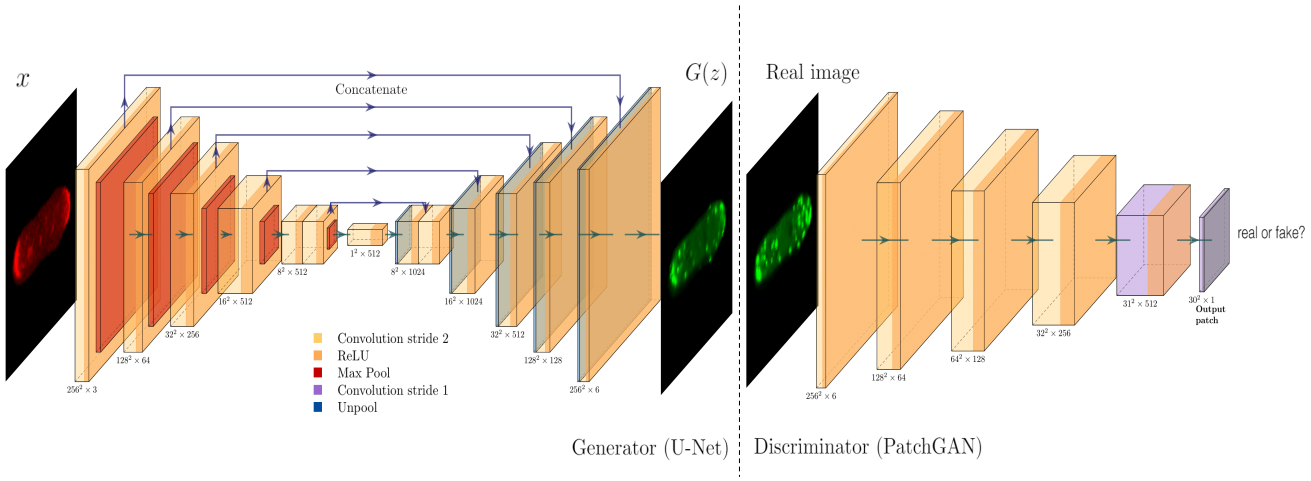

Figure 1: 2D pix2pix network schematic showing the U-net generator with skipped connections and PatchGAN discriminator with an output patch. Each operation and activation function applied at each layer of the network is highlighted in the figure.

encoder-decoder structure with concatenation between layer n and layer $N - n$ where N is the total number of layers [33]. The concatenation between the layers ensures that low-level information such as cell shape is kept in the generation of the output image and not lost in the bottleneck. This generates an output much closer to the true image in less iterations compared to a simple encoder-decoder generator. The PatchGAN discriminator classifies an $N \times N$ patch of an image as either real or fake and averages over the classifications of all patches to give a final classification for an image. This is faster compared to a traditional discriminator as the patch size can be varied and it does not have to classify the image pixel by pixel. For 2D pix2pix, the total network has 16 layers for images of 256×256 resolution. Objective function optimisation is carried out using mini-batch stochastic gradient descent with the Adam optimiser and a learning rate of 0.0002 for the first half of the epochs which then linearly decays until the final epoch [16]. Figure 1 shows the complete network and each function applied at each layer for a 2D image.

A model in the style of the pre-mentioned 2D pix2pix model which takes volume data is achieved by adding convolutions in the z-direction. Keeping with a U-net structured generator, each layer has an additional dimension and uses a 3D convolutional 4x4x4 kernel which moves over the volume, with stride length 2, accommodating for the additional dimension.

Due to memory being a limiting factor, a full 3D GAN taking 256x256x256x3 is infeasible with the available resources, so a reduced model taking 64x64x64x1 volumes with a single channel is considered, so an added downscaling of the volumes must take place.

These modifications result in a U-Net generator with 6 convolutional layers, with each layer utilising leaky ReLU, followed by 6 deconvolutional layers: each with ReLU, the first two layers utilises 50% dropout and the final layer using a sigmoid function. The discriminator adopts the same style modification, having 3 convolutional layers with batch normalisation and leaky ReLU, and the final layer uses a sigmoid function.

The total average training time for 2D pix2pix was ∼5 hours and ∼8 hours for 3D pix2pix on an NVIDIA Quadro RTX 6000. 2D pix2pix was implemented in PyTorch (1.4.0) and 3D pix2pix was implemented in TensorFlow (2.2.0).

3.3 Mock Data Generation

Before applying the pix2pix code on real data, 2D mock cells were generated that mimicked some of the artefacts observed in protein distributions within real *Dictyostelium* cells. The importance of this is to test the capabilities of the model before attempting to learn more complex pairwise relationships. Arbitrary 2D binary masks were generated using Gaussian noise and various filters were applied to the mask, submask and their edges to capture the distributions of observed travelling actin waves [32]. These waves are constitutional of the same proteins as macropinocytosis cups and resemble frustrated cup structures that do not protrude outwards. Three pairs of multichannel cells were generated, each with a red channel representing the LimEΔ protein marker and multiple green channels, MyoB, Coronin and Arp3, based on real observed distributions.

In three dimensions, binary volume masks were created through the addition of varying frequencies of Perlin noise [25]. This addition created a natural looking

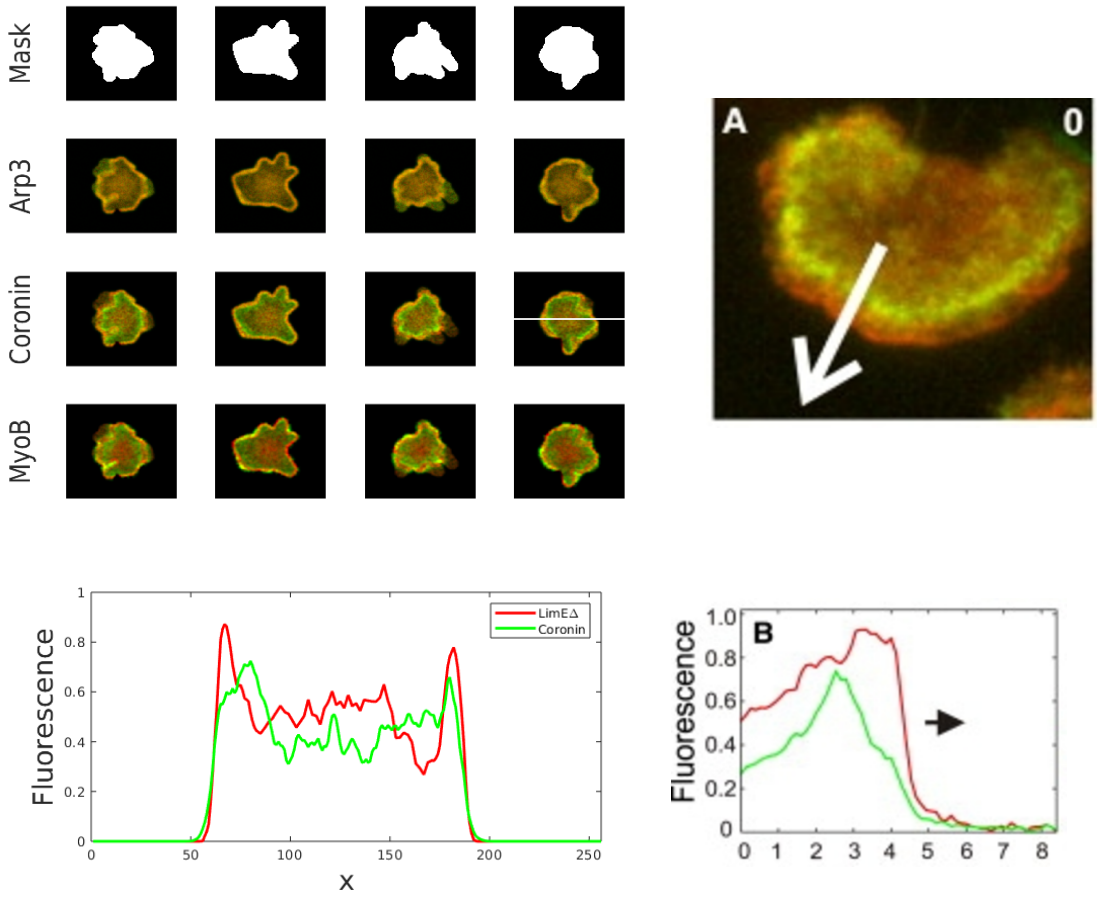

Figure 2: A table of mock proteins generated from four arbitrary binary mask shapes (Top Left). A TIRF image showing the localisation of coronin at the back of an actin wave from [32] (Top Right) and its distribution of the fluorescent intensities at the wave front (Bottom Right). Distribution of mock fluorescence of coronin and LimEΔ (Bottom Left) for the right-most mock coronin cell.

'cloud-like' texture across the entire space, rather than a cell-shaped mask. By combining this noise with an L2 weighting function centred on the centre of the space, this cloud begins to fade out as it spreads further from the centre of our 'cell'. Finally we apply a threshold to the space, resulting in a binary mask which appears visually similar to simpler *Dictyostelium* cells.

3D versions of the filters were used to generate mock MyoB and Coronin channels for these masks in a similar manner to the 2D distribution, however, the interior actin waves showed in Figure 2 were not replicated due to the inability to generate a sub-mask within the cell mask to then apply the filters to. Despite this, the mock cells present themselves as sufficient data to test the performance of the 3D GAN.

3.4 Evaluation Metrics

Evaluating generated images from a GAN is still very much an open problem with no consensus on the most appropriate metrics. The objective function of a GAN is intractable, therefore likelihood maximisation is very difficult. The most common GAN evaluation measures are carried out using generated samples and statistical sample tests to assess the quality of generated images [34]. Sample tests are very informative as perfect samples would imply zero KL-Divergence. All image quality metrics presented in this report are quantitative as qualitative evaluation can be misleading in high-dimensional images.

The first metric which is examined is the Manhattan (L1) distance between the generated images and the real images. This is the metric used in objective function optimisation during training in pix2pix. The generated images were only accepted if the L1 loss converged during training. However, this is the minimum requirement in generating realistic images and does not guarantee that the mapping has been learnt.

The inception score (IS) introduced by Salimans et al. [35] is currently the commonly used metric for GAN

evaluation. The IS is defined as:

$$IS(G) = \exp(\mathbb{E}_{\mathbf{x}}[D_{KL}(p(y|\mathbf{x})||p(y))]), \tag{2}$$

where $\mathbf{x}$ is a generated image sample, $p(y|\mathbf{x})$ is the conditional label distribution of samples estimated using a pre-trained Inception network [36] and $p(y)$ is the marginal class distribution. IS does not consider the real images and, therefore, can only measure image diversity but it has been shown to coincide well with human evaluation of image quality [37].

An emerging popular metric for GAN evaluation is the Fréchet Inception Distance (FID) proposed by Huesel et al. [21]. FID compares the distributions of the generated and real images by embedding the samples into the feature space of the last layer of the Inception V3 network. Both distributions are therefore modelled as multivariate Gaussians with estimates for both the mean and covariance. The distance between these two Gaussians is then given by the Wasserstein-2 distance:

$$\begin{aligned} FID((\mu_r, \Sigma_r), (\mu_g, \Sigma_g)) =& ||\mu_r - \mu_g||_2^2 \\ &+ Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}), \end{aligned} \tag{3}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the means and covariances of the real and generated samples respectively. This distance is then used to infer the quality of the generated images with a lower FID score implying better generated images. This metric assumes a Gaussian distribution, however, this can be a bad approximation of the data and the difference between image quality and image diversity is not considered.

A common method of assessing image quality of generated biological images is using Haralick's texture features proposed by Haralick et al.[22]. This involves computing the Gray-Level Co-occurrence Matrix (GLCM) of the foreground of an image where each entry, $p(i, j)$, is the probability that a pixel with value i is adjacent to a pixel with value j where i and j are image intensity. Haralick defined fourteen textural statistics derived from the GLCM, however, as carried out in the literature [12, 23], we focused on entropy, contrast, correlation and homogeneity. These statistics are defined as:

$$Entropy = -\sum_i \sum_j p(i,j) log(p(i,j)) \tag{4}$$

$$Contrast = \sum_i \sum_j (i-j)^2 p(i,j) \tag{5}$$

$$Correlation = \sum_i \sum_j p(i,j) \frac{(i-\mu_x)(j-\mu_y)}{\sigma_x \sigma_y} \tag{6}$$

$$Homogeneity = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j), \tag{7}$$

where $\mu_x, \sigma_x, \mu_y, \sigma_y$ are the means and standard deviations of $p(i)$ and $p(j)$ respectively. Entropy is a measure of randomness in the image, contrast is a measure of intensity between neighbouring pixels, correlation is a measure of linear dependency in the image and homogeneity is a measure of similarity between pixels. These statistics were calculated for both real and generated samples and then a non-parametric two-sample test (Kolmogorov-Smirnov) was carried out with significance level $\alpha = 0.01$. The null hypothesis is that both samples come from the same distribution. As multiple hypothesis testing is carried out, Bonferroni correction is applied which means that if $p \leq \alpha/24$, the null hypothesis is rejected. This can be a better GAN evaluation method than the FID score as more features are tested and not only the first two order moments as with FID.

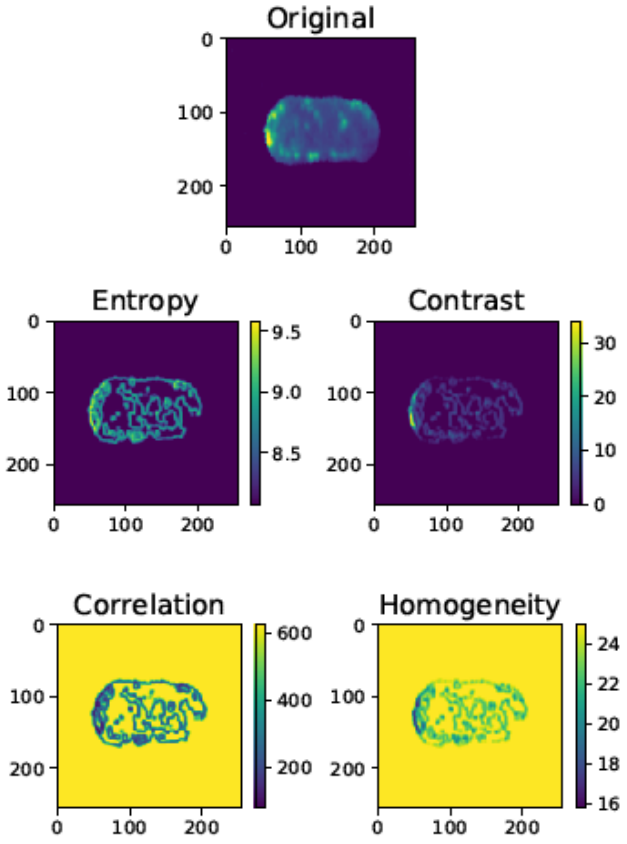

Figure 3: Haralick's texture features applied on a sample of a single channel real yeast cell. Entropy is largest and correlation is lowest between boundaries of differing pixel intensity. Contrast is largest in areas with the highest concentration of protein marker and homogeneity is lowest in these areas.

Image diversity in GAN generated samples is just as important as image quality. Mode collapse in GAN models occurs when the generator fails to produce a variety of samples and instead generates a few images regardless of input. Dropout reduces the likelihood of mode collapse, however as a sanity check, the birthday paradox test [38] is implemented and run on the generated samples. The birthday test involves taking a sample of size $\sqrt{S}$

from the generated dataset of size S and performing a measure of similarity which in our case we used a simple euclidean distance between two images. The most similar pairs are then extracted and visually inspected for near-similarity (as continuous distribution). If a near-duplicate exists, then the support size is estimated to be S^2 which if larger than the training set size, implies high diversity.

3.5 Using cell surface data

The surface of the *Dictyostelium* cell contains important information about where actin causes and creates visible protrusions on the cell when it undergoes macropinocytosis. The dataset we have available corresponding to this data is a time series for a single *Dictyostelium* cell consisting of 140 time steps, each containing a set of points alongside readings for PIP3 and LifeAct (actin) markers.

The main issue with using surface information is that the format is vastly different to the volume data of the same cell. In both two and three dimensions, the volume data obtained during imaging were of a standard form (after some necessary scaling and cropping), either an image of dimensions width × height, or a stack of images (width × height × depth). In contrast however, surface information is built up by creating a triangulation over a set of points scattered across the surface of the cell, which can have a varying number of points dependent on how accurate the triangulation aims to be. Therefore there is no standard definition of a surface that a GAN can accept as inputs – the number of surface points differs across datasets, as does the number of triangles.

As such, the data needs to be pre-processed into a standard format that can be used as input, regardless of the number of points or triangles on the surface.

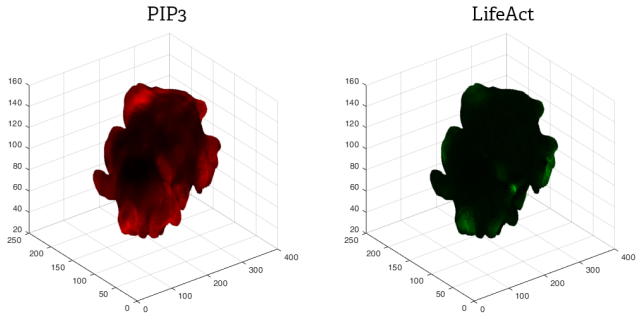

Figure 4: An example of the point cloud defining a *Dictyostelium* cell, showing both the PIP3 (left) and LifeAct (right) protein markers.

3.5.1 Pre-processing the surface

Due to the complex nature of the cell shapes there is no perfect strategy to convert cells to a standard format, however a sphere (or spheroid) is a close approximation to capture most of the input cells. This approximation should hold for most cell shapes apart from those which 'fold back on themselves' as this would cause some overlap when projecting onto a spherical surface.

We define the projection sphere as having a centre defined as the midpoint along each axis between the maximum and minimum coordinates. We then translate the surface to have this centre at $(0, 0, 0)$ and project the resulting set of coordinates to have radius 1 in spherical coordinate form.

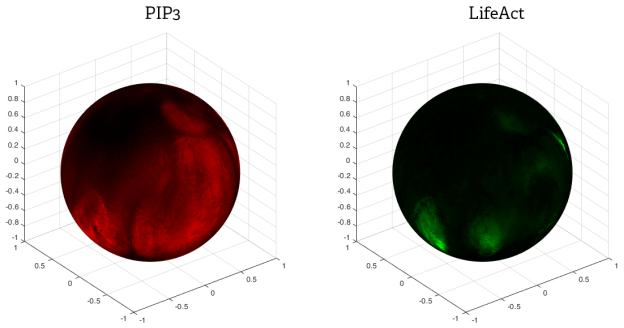

Figure 5: PIP3 (left) and LifeAct (right) protein marker readings mapped onto the surface of a sphere.

Next we generate a unit sphere consisting of 256×256 points which are evenly spaced around the sphere. Each of these then uses a nearest neighbour approximation to estimate the protein activation at that point. These 256×256 points can be viewed as a two-dimensional representation of the surface of the sphere. In this form they are capable of being used as training and testing input for a generative adversarial network.

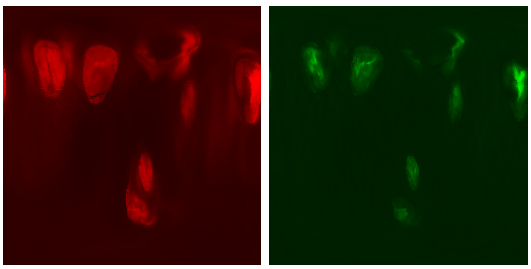

Figure 6: Two dimensional surface of the spherical approximation of the cell surface for PIP3 (left) and LifeAct (right). Contrast has been increased to improve visibility.

3.5.2 Data augmentation

Due to the low amount of data available, it is necessary to perform standard machine learning data augmentation techniques. In the case of 3D surface data, the data can be augmented through various rotations of the sphere about its centre point. This results in a larger and more varied dataset due to how the 2D surface is unwrapped from the sphere. This is necessary as the cells can be imaged in any rotation as they move and this behaviour needs to be captured when learning.

Through rotation alone, the dataset was expanded to 1540 pairs of images through randomly chosen rotations about the (x,y,z) axes. Extra transformations including reflections could have been applied to increase the amount of data available, however, this was deemed unnecessary due to the variation that rotation introduces. Translation was not employed due to this method using a centre point relative to the cell and as such would have no effect on the output surface data. Shearing and scaling are not suitable as they would distort the cell to biologically unrealistic shapes.

3.5.3 Mapping back onto a cell

Once a surface corresponding to the LifeAct marker has been generated by a trained GAN, it needs to be mapped back onto the original point cloud. This entire process will have a loss of precision due to the number of points within the point cloud being in excess of the number of points used in the spherical projection. The process of this mapping is to re-use the spherical coordinates obtained from projecting the entire point cloud onto a radius 1 sphere and interpolate from the generated surface. In this case we have used a simple nearest neighbour approach due to the computational inefficiency of performing a linear interpolation for the large amount of data points.

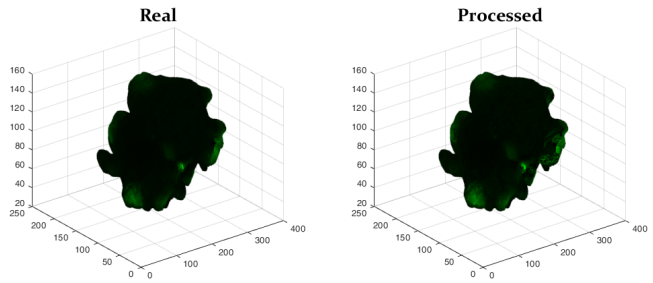

Figure 7: Point clouds containing the protein marker for actin: Original (left), after undergoing spherical transformations and mapped back onto the original points (right).

As seen in Figure 7, the loss of information when being sampled into a 256×256 surface image becomes apparent in the process of mapping back onto the cell. This is evident through the almost noise-like appearance on the rightmost protrusion, which no longer has the smooth appearance it should have. This downsampling is necessary due to the imbalance between the number of points in the original surface point cloud and the number of pixels within the output surface image. Within the point clouds, there are of the order of $100,000$ points, however the surface image has a dimension of $256 \times 256 = 65536$, resulting in the data being downsampled by at least a factor of 2. This combined with the non-uniform distribution of points when projected onto the sphere causes reasonable issues with the quality of the cell surface. This issue can be trivially overcome by increasing the number of points used in the surface projection, resulting in an increased size surface image, as even just increasing the image to be 512×512 would increase the number of pixels to be double that of the number of points. This would increase the accuracy at the expense of increasing the number of parameters to train in the neural networks.

4 Results and Discussion

4.1 Qualitative 2D results

4.1.1 Yeast cells

After concatenating the (red) reference marker image with the (green) protein marker image into a pair, 2000 image pairs made up the whole dataset. The training set contained 1600 pairs and the test set contained 400 pairs. There was no validation set as Isola et al.[16] claims that the hyperparameters that they have chosen are optimal with no requirement for hyperparameter tuning. Training was conducted using standard pix2pix architecture for 200 epochs with the L1 loss having converged.

The model output and target images are qualitatively very similar as seen in Figure 8. In real yeast cell images, regions of high concentration of Bgs4 (red reference marker) are correlated with regions of high Act1 (green marker) concentration. This relationship has been learnt by the model and can be seen between the reference images and the generated images. However, the model is seen to struggle mostly on input images that do not have localised regions of very high concentration such as the images on the third row of Figure 8. A possible explanation for the worse performance is that the training data does not contain enough images with a more spread out distribution of Bgs4.

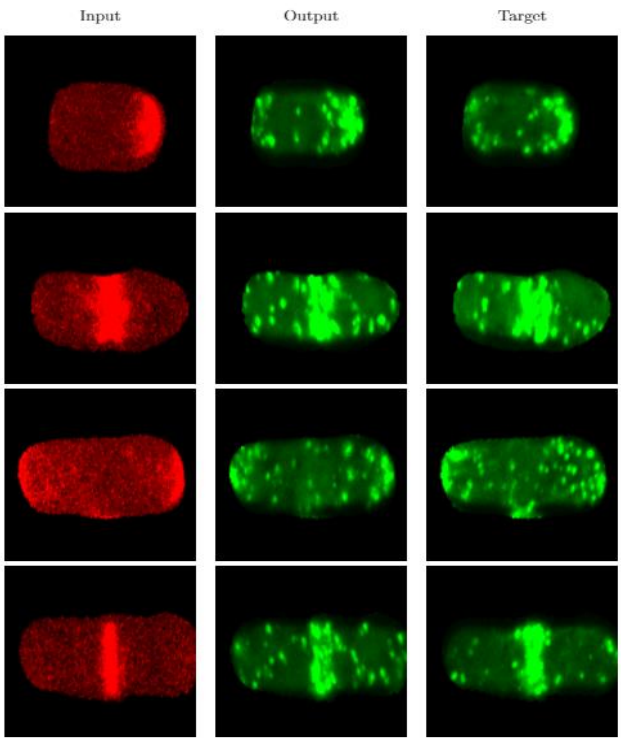

Figure 8: A selection of yeast cell test data from pix2pix trained for 200 epochs compared to the target.

4.1.2 Mock cells

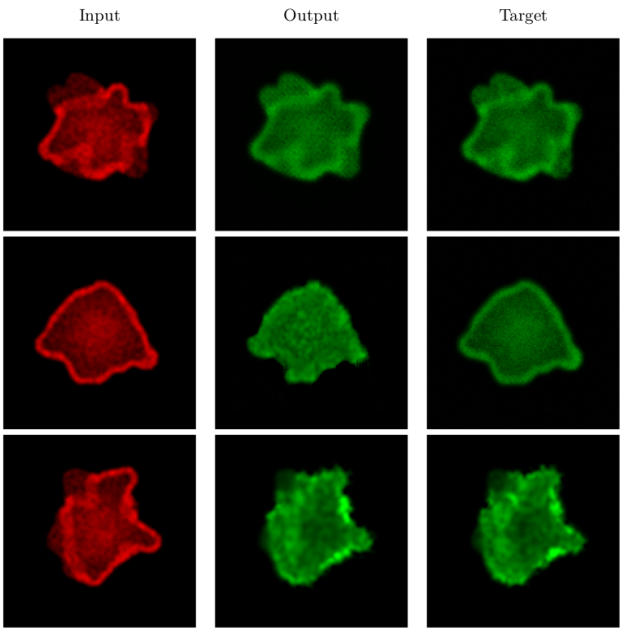

Figure 9: Example of mock input, output and target images for LimEΔ and Arp3 (top), Coronin (middle) and MyoB (bottom).

Three pix2pix models were trained on 500 mock cell images of LimEΔ red reference marker paired with Arp3, Coronin and MyoB green channels. Even with such a small training set, the generator was able to learn the distribution, Figure 9, presumably due to the low noise outside the cell, and the relatively simple relationships of the pairwise distributions within the generated cells. The trained model was tested on a set of 1000 test images.

4.1.3 *Dictyostelium* cells

2D (x-y) slices were extracted from 3 cells (hyperstacks) for a training set and 1 cell for the test set. A separate cell was used for the test set to have as much independence in the data from the training set as possible. The total dataset contained 18000 pairs with 15000 going in the training set and 3000 going in the test set. More training data was used for *Dictyostelium* cell data compared to yeast cell data as the relationship between red and green markers is more complicated for *Dictyostelium* cells. Standard pix2pix was used and the model was trained for 200 epochs with the L1 losses having converged.

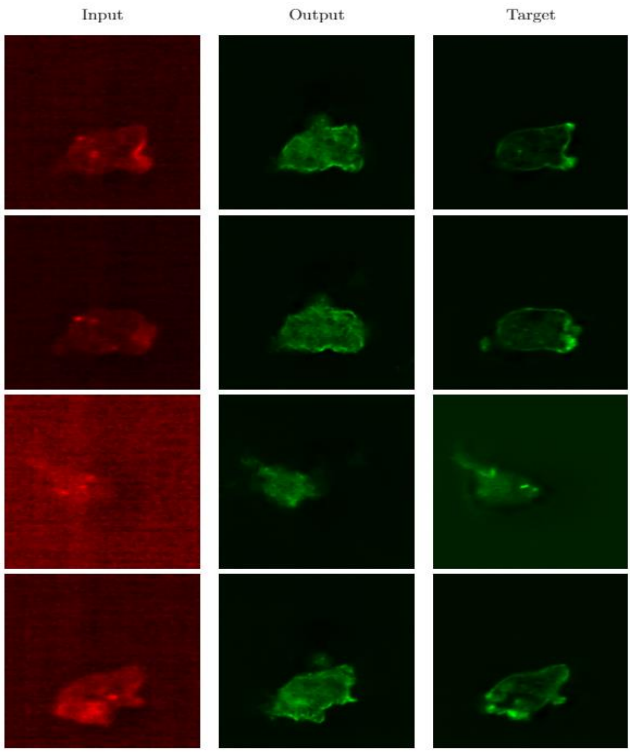

Figure 10: A set of normalised test samples of 2D *Dictyostelium* cell data trained on pix2pix on 15000 training data showing reference images (input), generated images (output) and real target images.

The mapping of *Dictyostelium* cell data appears to be

worse compared to yeast cell and mock cell data as shown between Figures 8, 9 and 10. More training data did not help with the more complicated mapping as there are clear qualitative differences between the model output and target images as seen in Figure 10. The distribution of the green marker protein is much more uniform and less localised in the generated image than the target image. This could be a result of some very noisy input data as seen in the first column of Figure 10 which occurs when the signal at cell edges is much lower which increases noise in the image.

4.1.4 Cell surfaces

As mentioned in the method, 1540 pairs of cell surfaces were output as a result of augmenting the 140 pairs obtained originally. These pairs were then split into training and test data with 1232 pairs (80%) being used for training and 308 pairs (20%) being used for testing.

This data was then used to train a generative adversarial network consisting of the standard pix2pix architecture for 250 epochs, the results of which were tested using the evaluation metrics documented previously. As these metrics operate primarily on image data, the results have been evaluated on the cell surface data directly generated from the GAN, not after it has been remapped onto the cell.

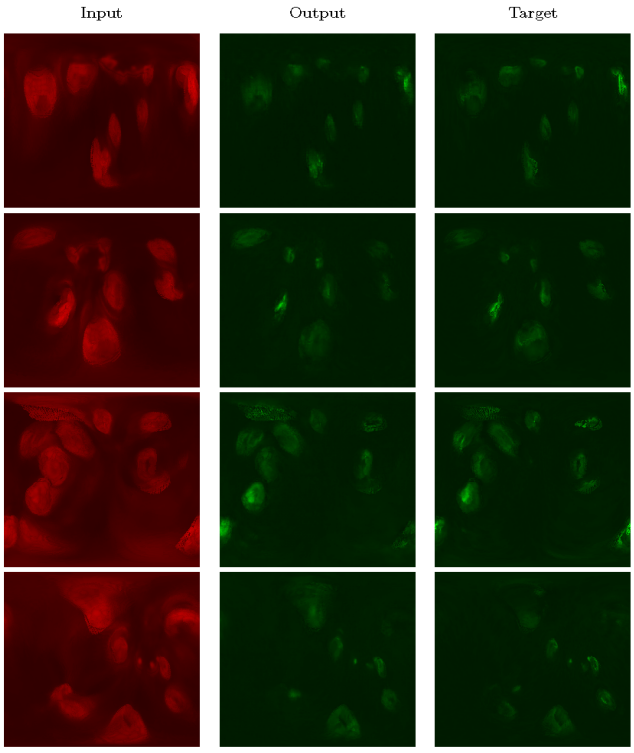

Figure 11: A selection of surface test data from pix2pix when trained on 1232 pairs of training data, tested on 308 pairs of data. Contrast boosted for visibility.

As visible in Figure 11, the mapping between PIP3 (red, column 1) and Actin (green, columns 2,3) is learnt to a reasonably accurate standard when studied qualitatively. Every area containing a high level of Actin is identified correctly, with only minor deviations at a pixel level. There are certain areas of the generated images which appear 'blurred', this is due to the the use of the L1 loss function. This is most noticeable in the first row of images, where the rightmost patch does not appear to be as sharp as in the target image.

4.2 Quantitative evaluation of 2D results

Each metric was calculated using all the test data from each 2D model. The IS does not require real data, however the FID score and Haralick's features were calculated using all test data and randomly sampled real data. For consistency, the test data and real data evaluation size were equal for the FID score and Haralick's features.

Dataset	IS	FID
Yeast cells	1.0477 ± 0.001	4.5654
MyoB mock cells	1.4162 ± 0.028	5.1470
Dictyostelium cells	1.0452 ± 0.009	17.0341
Cell surfaces	1.2567 ± 0.008	17.1280

Table 1: Evaluation metrics on 2D datasets showing the FID score and mean and standard deviation of the IS. For the IS, the larger the value the better the representation and for FID, the lower the value the better.

The most representative generated dataset of the real data is the MyoB mock cell dataset according to the IS, whereas, according to the FID score, the most representative generated dataset is the Yeast cell data as shown in Table 1. Both of these datasets were also qualitatively indistinguishable which supports our metric results. Surprisingly, the cell surface dataset scored much worse than the mock and yeast cell datasets which contradicts the qualitative results. However, a justification for this contradiction is that the variability of the surface cell dataset (1 cell used) was much lower compared to the other datasets which means the feature estimates were not accurately representative of the data and therefore, the metrics are inaccurate for cell surface data.

The IS values for all 2D datasets are very small compared to the IS scores seen in the literature [20, 37]. A score of 1 implies GAN collapse which has been showed to not have occurred in our models through the birthday paradox tests. The main reason for the low scores is a result of the classifier that is used when calculating the IS. The training data used to train the classifier has very little to no biological images which means the score will always be low irrespective of whether the generated

data is of high-quality. Therefore, the IS should not be used on its own for generated image evaluation.

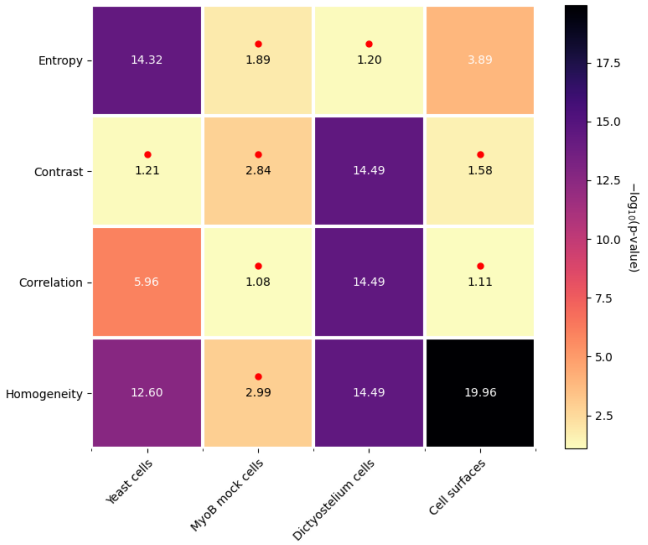

Figure 12: Heatmap showing the -log_{10}(p-value) for each of the four Haralick features and each 2D dataset. The red dots indicate for which features the null hypothesis is accepted at a significance of $\alpha = 0.01$ with Bonferroni correction. Larger values indicate greater variation between the real and generated features.

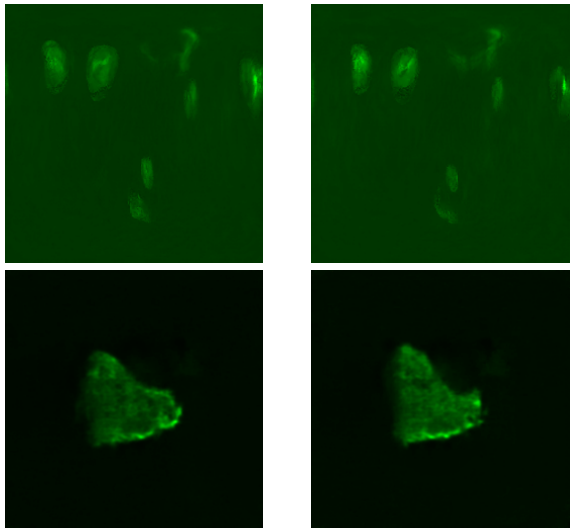

Figure 13: Near-duplicate pairs for the cell surface (1st row) and *Dictyostelium* cell (2nd row) datasets with euclidean distance of 5.569 and 8.225 respectively (lower means more similar).

Evaluation of the 2D datasets using Haralick's features highlighted in the method section is shown in Figure 12 with quantile-quantile (Q-Q) plots and box plots for each feature in Figure 18 (Appendix A). The statistical tests (K-S) imply that the generated MyoB mock dataset follows the same distribution as the real mock dataset for all features. This result is not surprising as the relationship between the red and green markers for mock data is much simpler to learn compared to a real biological process. The smallest p-value was for homogeneity in cell surfaces which is a result of mapping a surface onto a 2D image which distorts the original surface.

Examining only the Q-Q plots and box plots in Figure 18 (Appendix A), the only dataset with large deviations for each feature is the *Dictyostelium* cell dataset. Therefore, if only qualitatively accurate images to the human are desired, conducting a two-sample test on the features is not necessary and Q-Q plots and box plots provide sufficient information for evaluation.

The birthday paradox test was carried out five times on each dataset and the top ten most similar images were extracted for near-duplicate identification. The yeast cell and mock cell datasets did not contain any near-duplicates, therefore, mode collapse did not occur during training and the generated sample diversity is assumed to be high. A duplicate with probability $\geq 70\%$ was found in the generated cell surface dataset. As the sample size for the cell surface dataset was 30 images, the support size of the distribution is therefore $\approx 30^2$ which is smaller than the training set.The small support size of the cell surface generated images does not directly imply mode collapse in this case (as some variety exists), however, it is an indicator of the need to increase the diversity of the training set.

For the *Dictyostelium* cell dataset, a sample size of 150 images was used with a duplicate found with probability $\geq 50\%$. This means that the support size of the distribution is $\approx 150^2$ which is larger then the training set size of 15000.

4.3 3D Results

4.3.1 Qualitative 3D results

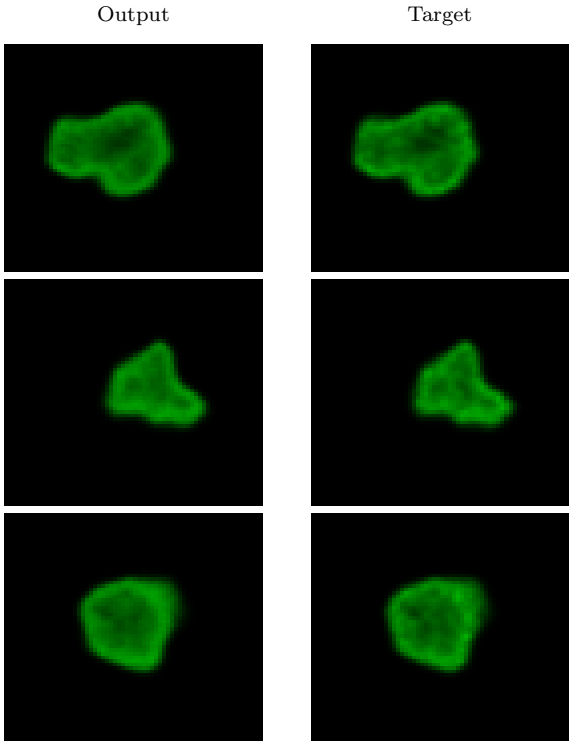

Figure 14: 3D GAN output compared to the target for mock Coronin. The output (left) is very close to the target distribution (right) despite exhibiting some additional blur. This result provides the confidence to move forward with the model.

The 3D GAN is tested on two datasets: 3D mock proteins for LimEΔ (red) and Coronin (green) and real volume data consisting of PIP3 and MyoB proteins dressed in red and green fluorescent proteins markers respectively. Firstly, 153 low complexity volumes of mock LimEΔ and Coronin protein were fed into the GAN and trained for 500 epochs. Due to their low complexity, the GAN had no trouble learning the relationship between the two generated proteins, shown in Figure 14. Despite the small training set, the positive result provides us with reassurance that the 3D GAN is capable of learning the pairwise distributions in volume data.

Secondly, the 3D GAN was applied to the real volume data extracted from multi-channel timeseries volume data of PIP3 and MyoB marked proteins stored in hyperstack format. Of the 720 available volumes, 48 were held out for testing a trained GAN on the remaining 672, which for 100 epochs took 7.5hrs. Due to the resizing of the data to account for the limiting factors, we must bear in mind that the outputs will be of a lower resolution to the raw data. Nevertheless, we can directly compare the output of a trained 3D GAN with the pseudo-3D output generated by a modified 2D GAN at the same resolution [12].

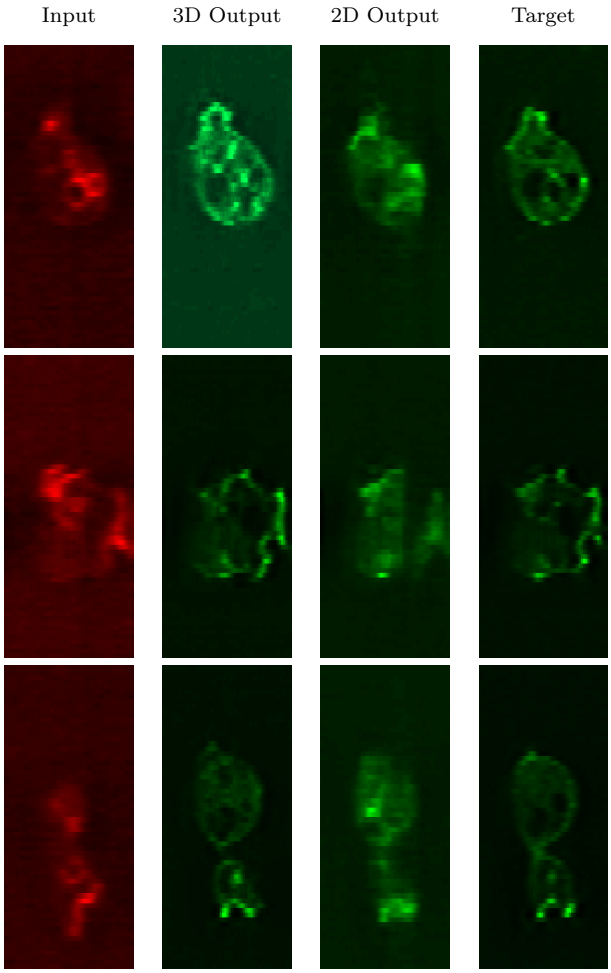

Figure 15: Cross sections of the input, 3D output, pseudo/2D output and target volume data along the zy-plane. As shown in 16, visually the 3D GAN outperforms the 2D GANs ability to produce informative pseudo-3D output.

Similar to the previous section, the 2D GAN generations fail to capture some of the key biological artefacts in macropinocytosis such as the well defined, highly curved areas of the cell membrane and intracellular areas corresponding to the nucleus and vesicles as seen in Figure 16. Such artefacts are presented more clearly in the 3D GAN output, so qualitatively we can conclude that the additional convolution in the z-direction improves the models ability to learn these complex distributions. For completion, Figure 15 shows the distributions along the zy-plane, demonstrating the poor approximation of pseudo-3D output along the direction of no convolution as a result of the 2D GAN.

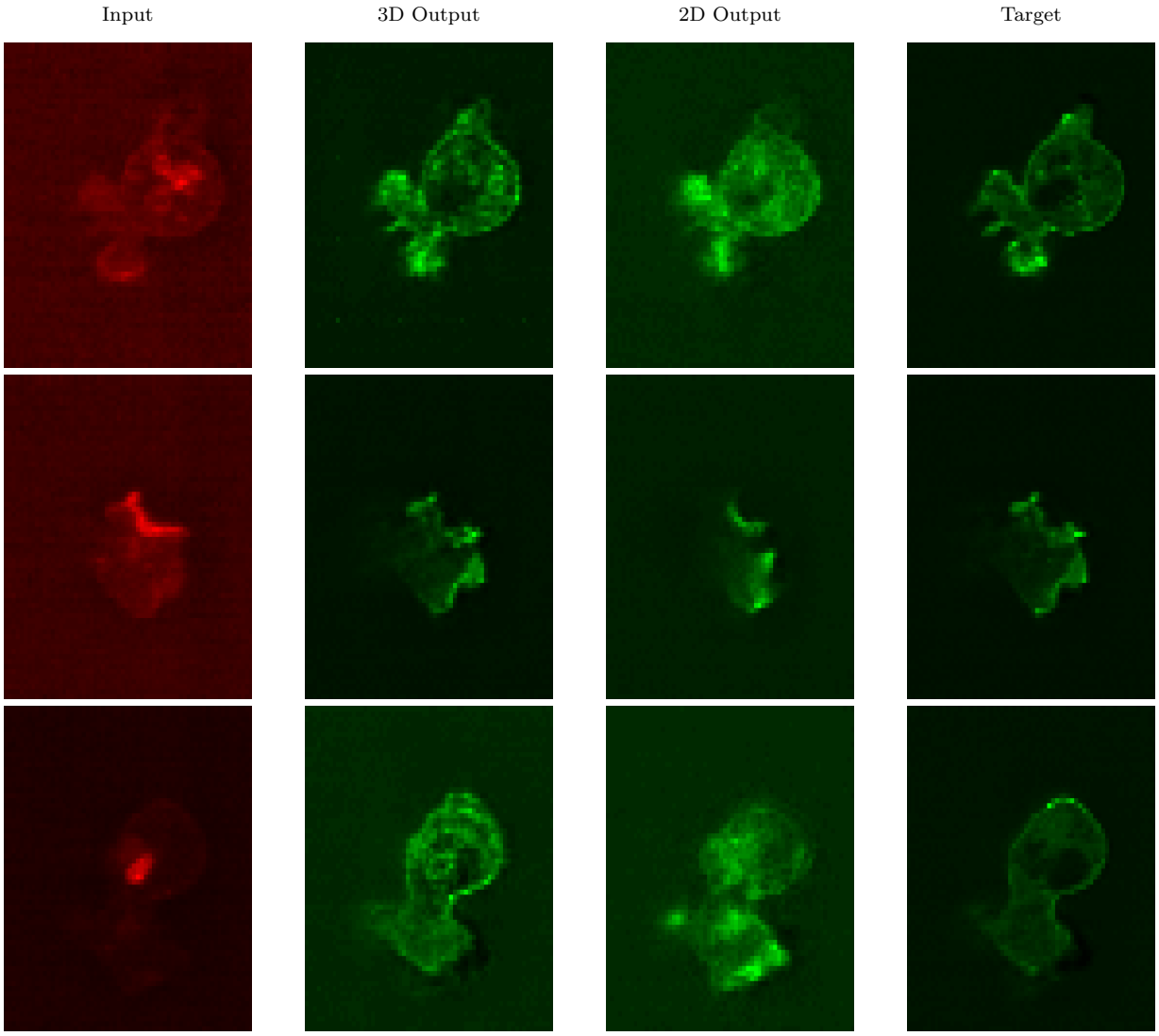

Figure 16: Example of cross sectional input along the xy-plane, which is fed into both the 3D and 2D GAN. The target distribution in the right column shows the green channel real data associated with the red input on the same row. Visually, the 3D output preserves the biological features such as intracellular structures and the sharp protruding membrane. In contrast, the 2D GAN output is less clear, dark areas within the cell where a vesicle or nucleus would be are not as well defined as the 3D output (Top and bottom row displays this well). Each image is normalised individually for visual purposes and not a true reflection of a true fluorescence comparison between images.

4.4 Quantitative evaluation of 3D results

Dataset	IS	FID
3D Mock cells	1.2408 ± 0.029	6.1000
z-y 2D cell slices	1.3455 ± 0.028	37.7600
x-y 2D cell slices	1.1322 ± 0.046	40.6975
z-y 3D cell slices	1.4443 ± 0.030	8.8719
x-y 3D cell slices	1.3473 ± 0.059	18.8399

Table 2: Evaluation metrics on 3D datasets and low resolution 2D dataset showing the FID score and mean and standard deviation of the IS.

We evaluated the 3D models by sampling 2D slices from the 3D cells for each dataset. The metrics that were applied for the 2D generated cells were applied to the extracted 2D slices from the 3D generated cells. It is immediately noticeable in Table 2 that the FID scores are much larger for the pseudo-3D results compared to the full 3D model and even the 2D scores from Table 1. This large difference in FID scores is a result of reducing the resolution of the images as the FID score is very sensitive to resolution. Both the IS and FID score imply that the full 3D GAN generated cells are more representative compared to pseudo-3D generated cells. 3D mock cell data has very good metric scores which, just like the 2D mock cells, are a result of the very simple distribution

that is easily learned by the GAN.

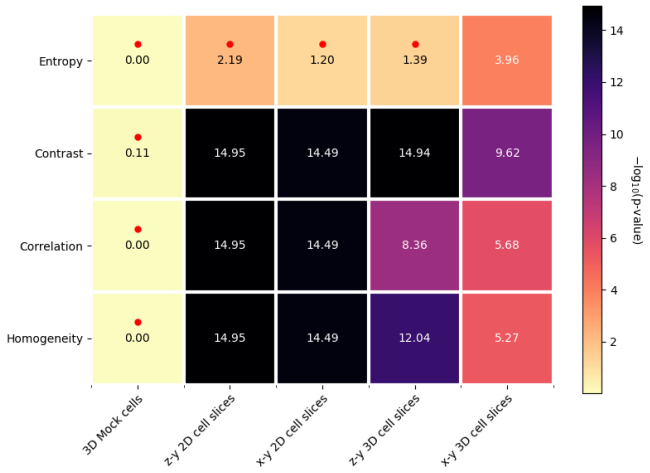

Figure 17: Heatmap showing the -log_{10}(p-value) for each of the four Haralick features and each 3D dataset and low resolution 2D dataset.

Haralick feature evaluation supports the metric values obtained for the 3D mock cell data as the null hypothesis is accepted for all features for the mock cells as shown in Figure 17. The feature results do not favour the 3D GAN as strongly as the metrics from Table 2, however, it is clear that the distribution of features for the 3D generated cells are much closer to the real distribution than the pseudo-3D results in both the x-y and z-y planes.

The birthday paradox test was carried out on the 3D generated cells with no near-duplicates which implies a large support size of the generated distribution and therefore, higher diversity compared to pseudo-3D generated cells. A limitation of considering 2D slices of 3D cells instead of the full 3D structure is that only small sections of cells are evaluated which can result in misleading metric results. However, these metrics are useful for gaining an understanding of the general quality of the generated distribution.

5 Conclusion

We have investigated three different approaches with varying levels of complexity in this paper, a network for learning the mapping between two proteins in two dimensional image data, using cell surface data projected into a standard form with a GAN, and a GAN capable of performing full three dimensional convolutions to learn a mapping for three dimensional volume data. Alongside this we investigated several evaluation metrics to understand how accurate our trained generative models were.

With the two dimensional methods involving yeast cells, 2D slices and cell surfaces, the qualitative results show a good similarity between the likeness of the generated images and the targets they were supposed to mimic. For both yeast cells and 2D slice data, the largest issues with the generative models came from issues with the input data, whereas for cell surfaces issues arose due to the process of downsampling. With the yeast cells, the main issue was predicting areas of localised regions of high concentrations of protein marker arising from a lack of this type of training data. With 2D slices however, the issues were due to noisy input data having an effect during training. In the cell surfaces, qualitative issues arose from the process of mapping the generated surfaces back onto their original cells, caused by the process of downsampling when projecting onto the spherical surface. As mentioned, however, this is understandable and can be fixed by increasing the dimensions of the surface image generated.

The 3D qualitative analysis showed that the full 3D GAN outperformed the 2D GAN by preserving many of the biological artefacts such as sharp protrusions of the cell membrane, the intracellular vesicles caused by macropinocytosis and the cell nucleus at low resolution. In addition, the resolution in the zy-plane was unaffected in the 3D GAN due to the convolutions in the z-direction, whereas the pseudo-3D volumes formed by the 2D GAN captured the structures along the zy-plane with very low resolution. We conclude that, qualitatively for this specific task, the more efficient pseudo-3D resultant green channel volumes were not a good substitute for those made by the 3D GAN.

The quantitative results using the varying evaluation metrics have shown us that generating representative biological data is very sensitive to data quality such as the resolution and variety of the training set. The quantitative evaluation supported the qualitative conclusion of the 3D GAN generating more representative samples compared to a pseudo-3D GAN with lower FID scores (8.87 compared to 37.76) and Haralick's feature two sample tests showing larger correlation between generated and real cell distributions. It has also been shown that the FID score and Haralick's texture feature analysis are viable and useful metrics for classifying whether a generated biological image is qualitatively representative of the true data.

5.1 Future Work

Due to the range of techniques and varying datasets we have used, there are several extensions that can be made to both our work and in the broader context of using generative adversarial networks with imaged cells.

5.1.1 Extension to multiple channels

One possible extension is to exploit the architecture of pix2pix in creating multi-channel images consisting of multiple proteins as currently we have been focusing in each case on a single input protein being mapped to a single output protein. This is not without its own issues however, as obtaining training data will still require the cells to be imaged with multiple markers which can be experimentally infeasible or even impossible.

5.1.2 Spherical architecture for surface mapping

In the surface mapping case, there are key limitations which may play a part in the accuracy of the results, the largest of which arises from the projection of the data onto the surface of a sphere. With ordinary volume data, there is a Euclidean relationship between the pixels (or voxels in three dimensions), however, this projection does not maintain that relationship, with distortions being present around the poles and equatorial regions of the sphere.

One potential fix for this would be to tweak the architecture of the GAN (pix2pix in our case) to take the spherical nature of the data into account, rather than its default 'planar' architecture. Multiple methods related to this exist, for example image recognition on omnidirectional images uses a specialised kernel which wraps around the sphere in a regular manner [39].

This approach can operate hand in hand with increasing the quality of the cell surface estimation. The current method projects surfaces down to (256×256) images, which is a large downsampling from the original number of points present within the data. Increasing the number of sampled points does not need to happen uniformly, due to how they are spread over the surface of the sphere. Instead, increasing the number of points that represent the equatorial regions should be prioritised.

5.1.3 Further evaluation

Further research and analysis into many other evaluation metrics is essential for robust comparison and classification of generative models. A natural extension to the quantitative evaluation carried out in this report is applying more metrics for comparison such as the Classifier Two Sample Test (C2ST) [15] and Maximum Mean Discrepancy (MMD)[40] which would allow comparison with other GAN models.

The evaluation can also be extended for 3D GANs by considering complete 3D cells instead of 2D slices using an inverted generator and computing the FID scores. Another potential metric for 3D evaluation is the Jensen-Shannon Divergence in the 3D space which has been used to evaluate 3D GANs using point cloud data[31].

Acknowledgements

Supervised by Till Bretschneider with assistance from Josiah Lutton. Data obtained in collaboration with our industrial partner 3i.

Code

All of the code developed for this project can be found on GitHub: `https://github.com/MA932-GANS`.

References

[1] R. Parton, "Clathrin independent endocytosis," in *Encyclopedia of Cell Biology*, R. A. Bradshaw and P. D. Stahl, Eds. Waltham: Academic Press, 2016, pp. 394 – 400.

[2] J. Swanson and S. Yoshida, "Macropinocytosis," in *Encyclopedia of Cell Biology*, R. A. Bradshaw and P. D. Stahl, Eds. Waltham: Academic Press, 2016, pp. 758 – 765.

[3] C. V. Halder, E. M. Fonseca, A. V. de S. Faria, and S. P. Clerici, "Chapter 11 - extracellular vesicles as a recipe for design smart drug delivery systems for cancer therapy," in *Drug Targeting and Stimuli Sensitive Drug Delivery Systems*, A. M. Grumezescu, Ed. William Andrew Publishing, 2018, pp. 411 – 445.

[4] G. L. Ryan, N. Watanabe, and D. Vavylonis, "Image analysis tools to quantify cell shape and protein dynamics near the leading edge," *Cell structure and function*, vol. 38, no. 1, pp. 1–7, 2013.

[5] M. V. D'Ambrosio and R. D. Vale, "A whole genome rnai screen of drosophila s2 cell spreading performed using automated computational image analysis," *Journal of Cell Biology*, vol. 191, no. 3, pp. 471–478, 2010.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

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

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[9] G. R. Johnson, R. M. Donovan-Maiye, and M. M. Maleckar, "Generative modeling with conditional autoencoders: Building an integrated cell," *arXiv preprint arXiv:1705.00092*, 2017.

[10] T. Ngo Trong, J. Mehtonen, G. González, R. Kramer, V. Hautamäki, and M. Heinäniemi, "Semisupervised generative autoencoder for single-cell data," *Journal of Computational Biology*, 2019.

[11] A. Osokin, A. Chessel, R. E. Carazo Salas, and F. Vaggi, "Gans for biological image synthesis," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] P. Baniukiewicz, E. J. Lutton, S. Collier, and T. Bretschneider, "Generative adversarial networks for augmenting training data of microscopic cell images," *Frontiers in Computer Science*, vol. 1, p. 10, 2019.

[13] P. Goldsborough, N. Pawlowski, J. C. Caicedo, S. Singh, and A. E. Carpenter, "Cytogan: Generative modeling of cell images," *bioRxiv*, 2017.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[15] D. Lopez-Paz and M. Oquab, "Revisiting classifier two-sample tests," *arXiv preprint arXiv:1610.06545*, 2016.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 213–229.

[18] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[19] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 247–12 258.

[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in neural information processing systems*, 2017, pp. 6626–6637.

[22] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[23] D. V. Sorokin, I. Peterlik, V. Ulman, D. Svoboda, T. Nečasová, K. Morgaenko, L. Eiselleová, L. Tesařová, and M. Maška, "Filogen: a model-based generator of synthetic 3-d time-lapse sequences of single motile cells with growing and branching filopodia," *IEEE transactions on medical imaging*, vol. 37, no. 12, pp. 2630–2641, 2018.

[24] T. Necasova and D. Svoboda, "Visual and quantitative comparison of real and simulated biomedical image data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[25] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[26] W. Dong, X. Zhang, and C. Zhang, "Generation of cloud image based on perlin noise," in *2010 International Conference on Multimedia Communications*. IEEE, 2010, pp. 61–63.

[27] M. Dustler, P. Bakic, H. Petersson, P. Timberg, A. Tingberg, and S. Zackrisson, "Application of the fractal perlin noise algorithm for the generation of simulated breast tissue," in *Medical Imaging 2015: Physics of Medical Imaging*, vol. 9412. International Society for Optics and Photonics, 2015, p. 94123E.

[28] S. D. Wiesner D., Nečasová T., "On generative modeling of cell shape using 3d gans." in *Image Analysis and Processing – ICIAP 2019. ICIAP 2019. Lecture Notes in Computer Science*, S. C. L. O. M. S. S. N. Ricci E, Rota Bulò S, Ed., 2019, vol. 11752, pp. 672–682.

[29] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum, "Marrnet: 3d shape reconstruction via 2.5d sketches," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 540–550.

[30] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum, "Learning shape priors for single-view 3d completion and reconstruction,"

in *The European Conference on Computer Vision (ECCV)*, September 2018.

[31] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," 2017.

[32] T. Bretschneider, K. Anderson, M. Ecke, A. Müller-Taubenberger, B. Schroth-Diez, H. C. Ishikawa-Ankerhold, and G. Gerisch, "The three-dimensional dynamics of actin waves, a model of cytoskeletal self-organization," *Biophysical Journal*, vol. 96, no. 7, pp. 2888 – 2900, 2009.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[34] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[37] S. Barratt and R. Sharma, "A note on the inception score," *arXiv preprint arXiv:1801.01973*, 2018.

[38] S. Arora and Y. Zhang, "Do gans actually learn the distribution? an empirical study," *arXiv preprint arXiv:1706.08224*, 2017.

[39] B. Coors, A. Paul Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 518–533.

[40] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

A Haralick's texture feature plots

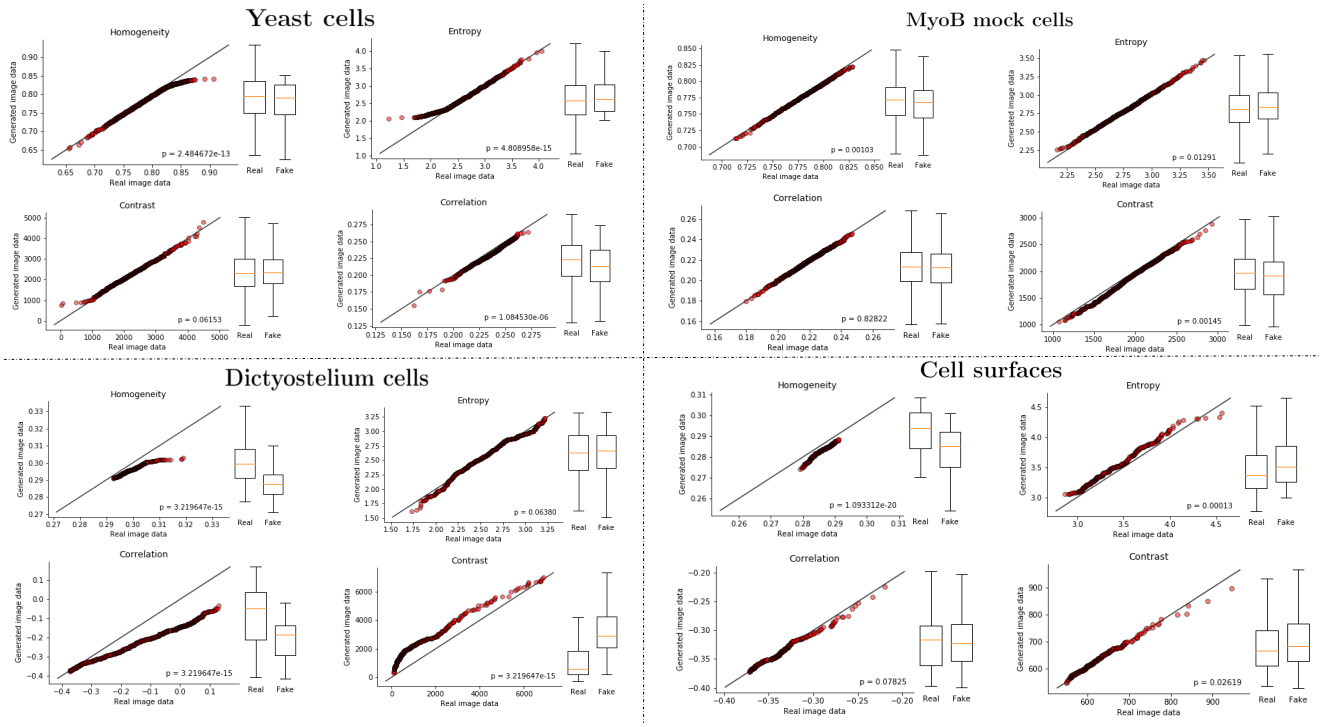

Figure 18: Quantile-Quantile plots and box plots with p-values for each Haralick feature for each 2D dataset.

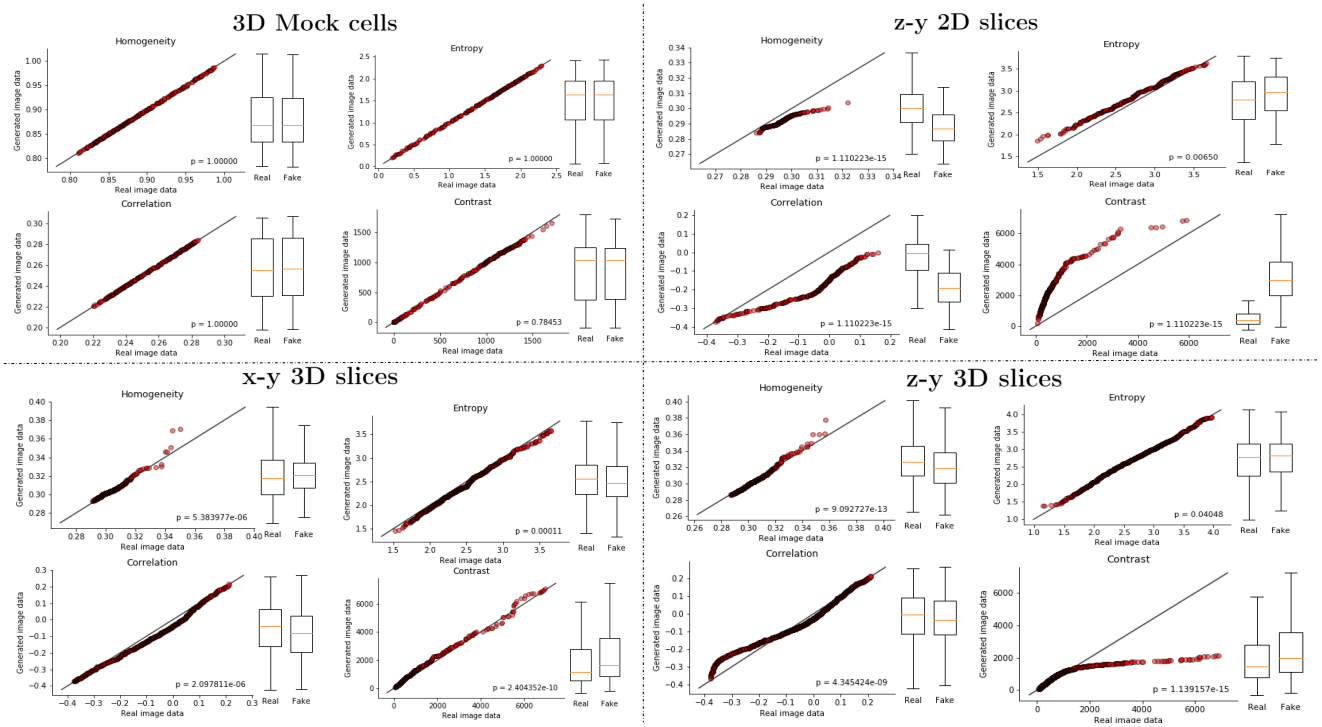

Figure 19: Quantile-Quantile plots and box plots with p-values for each Haralick feature for each 3D dataset and low resolution 2D dataset.