Morphology Based Cell Classification

Unsupervised Machine Learning Approach

by

Dhananjay Bhaskar

B.Sc., The University of British Columbia, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2017

Abstract

Individual cells adapt their morphology as a function of their differentiation status and in response to environmental cues and selective pressures. While it known that the great majority of these cues and pressures are mediated by changes in intracellular signal transduction, the precise regulatory mechanisms that govern cell shape, size and polarity are not well understood. Systematic investigation of cell morphology involves experimentally perturbing biochemical pathways and observing changes in phenotype. In order to facilitate this work, experimental biologists need software capable of analyzing a large number of microscopic images to classify cells and recognize cell types. Furthermore, automatic cell classification enables pathologists to rapidly diagnose diseases like leukemia that are marked by cell shape deformation.

This thesis describes a methodology to identify cells in microscopy images and compute quantitative descriptors that characterize their morphology. Phase-contrast microscopy data is used for the purpose of demonstration. Cells are identified with minimal user input using advanced image segmentation methods. Features (e.g. area, perimeter, curvature, circularity, convexity, etc.) are extracted from segmented cell boundary to quantify cell morphology. Correlated features are combined to reduce dimensionality and the resulting feature set is clustered to identify distinct cell morphologies. Clustering results obtained from different combinations of features are compared to identify a minimal set of features without compromising classification accuracy.

Preface

This is the original and unpublished work of Dhananjay Bhaskar. No ethics approval was required for this work. The *in-vitro* pancreatic carcinoma images (MIA PaCa-2 human cell line) used to demonstrate the methodology were acquired by Pamela Dean, PhD student working with Dr. Calvin Roskelley at The University of British Columbia (UBC), using phase-contrast microscopy.

NSERC USRA Project

The feature extraction section in Chapter 2 is based on work done in collaboration with Darrick Lee as part of an Undergraduate Summer Research Award (USRA) project from May to August 2016. Darrick implemented code to compute circular, elliptical, polygonal and cubic spline fits to cell boundary.

URO REX Mentorship Program

Dhananjay supervised undergraduate students MoHan Zhang and Cindy Tan from November 2016 to March 2017 as part of the Research Experience Program (REX) organized by Undergraduate Research Opportunities (URO), a student-run organization at UBC. MoHan improved and extended the feature computation work performed by Darrick Lee. MoHan also implemented code to compute rectangular fits to cell boundary with guidance from Dhananjay. Cindy investigated the use of silhouette score analysis and gap statistic to determine number of clusters in synthetically generated data. Partial results from Cindy's project are presented in the unsupervised classification section of Chapter 2.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I am greatly indebted to Dr. Leah Edelstein-Keshet for giving me the opportunity to pursue undergraduate and graduate research at UBC under her supervision. In addition to providing generous funding and unconditional support at a moment's notice, Dr. Keshet is largely responsible for my enthusiasm to pursue interdisciplinary research as a career. Under her guidance, I was able to pursue multiple independent projects of my own volition and acquire knowledge in fields as diverse as microscopy image processing, machine learning, mathematical modeling and multi-scale computational simulation of cell scale biology. Furthermore, Dr. Keshet introduced me to leading researchers in mathematical and computational biology by encouraging me to attend training workshops, present at conferences and collaborate extensively. The research reported here was supported by NSERC Discovery Grant (RGPIN-41870) awarded to Dr. Keshet.

I thank my mentors Dr. James Feng and Dr. Calvin Roskelley for their advice and constructive critique. Dr. Keshet and Dr. Feng organized joint research group meetings where I presented my work and received valuable feedback. Dr. Roskelley provided biological perspective and experimental data for multiple research projects. I thank the graduate chair, Dr. Daniel Coombs, for his valuable time spent reviewing the final draft.

I owe many thanks to my peers for their help and kind words of encouragement. Cole Zmurchok and Hildur Knútsdóttir are inspiring role models who invested their time to show me the ropes. At UBC, I had the incredible opportunity to work with talented undergraduate students: Eviatar Bach (Summer 2015 USRA), Darrick Lee (Summer 2016 USRA), MoHan Zhang and Cindy Tan (Fall and Winter 2016, URO REX Program), who motivated me to take on ambitious projects. Finally, I am grateful to my family and friends (Aditya Mandapati, Claire Guerrier, Lydia Lin and Michael Irvine) for their enduring support that has been instrumental in my success.

Dhananjay Bhaskar

Chapter 1

Introduction

Modern medicine and high throughput quantitative biology have fueled demand for computational methods that enable practitioners to analyze large quantities of data in an efficient manner. Data science and data mining can be used to inform experimental design, test diagnostic protocols, make predictions, verify and generate new hypotheses. Historically, the bioinformatics community has been at the forefront of developing tools that leverage advancements in parallel processing, GPU computing and machine learning to process 'omics' (e.g. genomics, proteomics, metabolomics) data. Increasingly, methods for extracting useful information from microscopic data are receiving greater attention [4]. From a modeling perspective, quantification of microscopic data is crucial to understand cell and tissue scale biology, including developmental, vascular and cancer biology. Translationally, availability of computational tools to process microscopic images is necessary for automating diagnoses and generating patient-specific treatments.

This thesis describes a methodology for identification of cells from phase-contrast microscopy images, quantification of cell geometry and morphology-based cell classification. The implementation of this methodology, resulting in an image processing and machine learning pipeline that operates semi-automatically (with minimal manual intervention), is suitable for analyzing a large number of images. Since the methodology does not rely on labeling cells with biomarkers, it can be easily adapted to work with confocal or fluorescent images. However, to illustrate how the method works, *in-vitro* phase-contrast images of pancreatic carcinoma cells (MIA PaCa-2 cell line) are used in this thesis. The MIA PaCa-2 cell line exhibits a number of different morphologies in monolayer culture. Therefore, it is ideal for testing, developing and benchmarking new analytical tools designed for two dimensional image analysis, cell recognition and classification. Phase microscopy was chosen for demonstration purposes because phase images are particularly challenging to segment [39].

The methodology described herein has numerous applications. The precise mechanisms for regulation of cell shape and size are not well understood. Such methodology can be used as an image-based screening procedure to determine whether an experimental perturbation (e.g. treatment with a chemical compound, small interfering RNA or genetic manipulation) leads to changes in cell morphology [35]. It can be used to detect outlier cell morphologies, i.e. cells that appear different from a control or reference group. Therefore, the methodology constitutes a systematic approach for scientific investigation of cell morphology. In a medical context, ability to identify and count different cell types from biological samples (e.g. blood, biopsy) can be used for diagnoses, estimation of risk and prognosis.

1.1 Literature Review

Manual classification of cells using computerized image analysis and quantification of morphological descriptors was well established before classification using machine-learning techniques became the norm. For instance, Merson-Davies and Odds classified *Candida albicans* cells into spherical, pseudohyphae (elongated) and true hyphae by computing morphology index (Mi) of each yeast cell using maximum length (l), maximum diameter (d) and septal diameter (s) obtained from image analysis [23]. They found that $\text{Mi} = ls/d^2$ is correlated with the content of chitin (a cell wall component) in cells and can be reliably used in place of subjective descriptions to identify cell morphology.

Machine learning techniques can be broadly categorized into two groups: supervised learning and unsupervised learning. In the context of using machine learning to classify data, supervised classification refers to techniques that require training data (i.e. subset of input data for which the classification outcome is known *a priori*) in order to "teach" a classification algorithm to recognize patterns in input data and get the desired output. In contrast, unsupervised classification does not require any training data, instead the input data is clustered or organized into different classes based on the inherent properties of the data. For the purpose of cell classification, input data is typically a randomized list of feature vectors, where each feature vector is an ordered set of "features" (sometimes referred to as "descriptors") that quantify the morphology of a cell. The terms " features" and "descriptors" are used interchangeably throughout this thesis. The remainder of this section describes recent work (2013 onwards) in the field of morphology based

cell classification and its applications.

A significant proportion of recent cell classification literature is devoted to diagnosis of leukemia. Putzu and Di Ruberto segmented nuclei of lymphocytes from blood samples in order to identify patients with acute lymphoblastic leukemia (ALL) [31]. They computed 30 shape descriptors, 4 color descriptors and 16 texture descriptors to quantify irregularities in nuclear shape and used a Support Vector Machine (SVM, a supervised learning method) to perform supervised binary classification. Textural descriptors were calculated for 0, 45, 90 and 135 degree rotations of the nucleus to maintain rotational invariance. Patients with ALL were identified with 0.25 percent mis-classification rate. The technique for segmenting nuclei using watershed algorithm described in this paper is also part of the methodology presented in this thesis.

Amin et al. further developed ALL classification methodology by computing a richer set of 77 geometrical and statistical features from blood and bone marrow smears to further categorize cell nuclei into 3 subtypes: L1, L2 and L3, based on the French-American-British (FAB) classification system [2]. Soon thereafter, Reta et al. classified bone marrow images to distinguish between families of acute leukemia (ALL and AML), subtypes of ALL (L1 and L2) and subtypes of AML (M2, M3 and M5) using a variety of classifiers [33].

Chankong et al. proposed a method for automatic cervical cell segmentation and classification using single-cell images obtained from a Pap test [7]. Nuclear and cytoplasmic regions of cells were separated computationally. Cells were classified into normal, low grade squamous intraepithelial lesion (LSIL), high grade squamous intraepithelial lesion (HSIL), and squamous cell carcinoma (SCC) phenotypic categories (in increasing order of malignancy) with over 95% accuracy. The authors computed six features to quantify the shape and coarseness of the nucleus, including some dimensionless shape factors that are described in the feature extraction section of Chapter 2. The feature vector also included three descriptors of overall cell shape and the size of the nucleus in relation to cell size. The authors compared results obtained from various supervised classifiers. These include K-nearest neighbors (KNN), artificial neural network (ANN) and SVM, not reviewed in this thesis.

Nosaka and Fukui classified fluorescence staining patterns of human epithelial type 2 (HEp-2) cells into six categories in order to perform automatic antinuclear antibody (ANA) analysis [26]. They trained a SVM classifier using features computed from multiple size-scaling and rotations of each cell image. The classifier outperformed other methods using the HEp-2 images data set created by the Mivia Lab at the University of Salermo.

Nanni et al. developed a methodology to classify phase-contrast microscopy images of retinal pigment epithelial (RPE) cells (derived from human pluripotent stem cell) using ensembles of cell texture descriptors [25]. Cells were classified into three maturation stages to asses suitability for implantation or *in-vitro* use. The methodology is generally applicable to a variety of biological image classification problems. However, it requires availability of an annotated data set to train SVM classifiers.

As co-culture systems gain popularity in the study of interactions between different cell types, automatic identification of distinct morphologies and cell types in co-culture microscopy is becoming more relevant. Logan et al. developed a pixel-based learning methodology (where pixel intensities of cell images are used as features) that can accurately identify multiple fluorescent morphologies [20]. Although it requires customized tuning for each cell type with distinct morphology, the authors demonstrated that the method can segment and count hepatocytes and fibroblasts in an unsupervised manner, without requiring explicit feature computation.

Ahonen et al. used unsupervised clustering methods to classify simulated tumor spheroids and images of PC3 human prostate cancer spheroids. They used geometrical features calculated from ellipse fitting, boundary features obtained from principal curve fitting and texture features comprised of local sample moments and local binary patterns to classify tumors using a clustering algorithm [1]. The complete feature vector consisted of 193 shape-based descriptors and 178 texture-based descriptors. After dimensionality reduction using principal component analysis (PCA), the authors identified 4 clusters using only geometrical features, 4 clusters using only boundary features and 3 clusters using only texture features. The four distinct clusters correspond to smooth spherical, spherical with rough borders, spherical with appendages and highly irregular phenotypes.

Finally, it should be noted that morphological classification of neurons presents a distinct set of challenges and idiosyncrasies. According to a recent review, supervised methods outperform unsupervised clustering methods despite the availability of large amounts of data due to lack of consensus in class delineation, sensitivity to algorithm parameters, limitations in feature extraction and lack of robustness in cluster identification [41].

1.2 Research Objective

The primary goal of the research described in this thesis is to develop a methodology for unsupervised classification of cells based solely on morphology-based features. While supervised classifiers have been shown to perform well, they require annotated input data for training and perform poorly if input data is too dissimilar to training data. In practical applications, annotated data is typically not available and generally requires significant time and effort to acquire. The insistence on using only morphological descriptors (called label-free classification) serves to keep the methodology generally applicable and free from limitations of labeling techniques or adverse effects of staining reagents. To facilitate development of the methodology and for the purpose of validation, phase-contrast images of pancreatic carcinoma cells (MIA PaCa-2) are used, although any microscopic data with reasonable spatial resolution and heterogeneity in cell shape would suffice. Current label-free classification methodologies mostly rely only on qualitatively singular type of features and cannot achieve multi-class classification [9]. Conversely, some methodologies that compute multiple feature descriptors cannot process high-throughput data due to computational complexity of feature extraction. Therefore, an objective of this work is to identify a minimal set of feature descriptors that can classify cells without compromising on the outcome.

1.3 Thesis Overview

The thesis assumes some background knowledge in image processing and machine learning. An attempt is made to define terminology and introduce concepts without detracting from the main content of the research. The thesis is organized into five chapters. Their contents are summarized below:

Chapter 1 provides an introduction to the research problem and a brief summary of recent literature in this field.

Chapter 2 describes the methodology used to perform image segmentation, feature extraction, dimensionality reduction and cluster identification. Together, the techniques described in this chapter form the basis of an unsupervised cell classification pipeline.

Chapter 3 illustrates a typical application of the methodology using the MIA PaCa-2 pancreatic cancer data set provided by the Roskelley Lab at UBC.

Chapter 4 summarizes the advantages and limitations of the methodology compared to other existing approaches.

Chapter 5 identifies areas for improvement and describes future work.

Appendix A describes classification results obtained by using t-SNE instead of PCA for dimensionality reduction.

Chapter 2

Methodology

This chapter describes a methodology for unsupervised classification of cells from phase contrast microscopy images. Typically, the machine learning algorithm is embedded into a processing pipeline that converts microscopy images into numerical data corresponding to individual cells [35]. The pipeline consists of image processing, feature extraction, dimensionality reduction, classification and validation steps, which are described in detail below.

The first phase of image processing is to enable separation of foreground and background by removing artifacts, reducing noise and compensating for uneven illumination. Subsequently, image segmentation methods (techniques that divide an image into regions of interest) are used to identify cells amongst the foreground pixels. The choice of segmentation algorithm depends on the image and cell type. No single algorithm is capable of identifying cells from multiple image sources. Often, manual tuning of algorithm parameters is required to achieve optimal performance.

Once cells are segmented, quantifiable descriptors of cell shape and size are computed. These are referred to as feature vectors and they form the basis for distinguishing between cells using a classification algorithm. Feature vectors are normalized to have zero mean and standard deviation of unity in order to prevent discrimination between features due to difference in their magnitudes or signs. The performance of the classifier depends on the quality of segmentation and accuracy of features. Two cells can be distinguished (i.e. assigned different labels by the classifier) if their feature vectors differ to a significant degree. Furthermore, in order to identify a certain morphology, one or more features must capture unique characteristics of that morphology.

Most widely used classification algorithms that rely on computation of distance metric between features tend to perform well for low dimensional

feature vectors. Dimensionality reduction techniques convert high dimensional feature vectors to low dimensional vectors by identifying correlations and combining multiple features in a manner that maximizes the variance between projections of the data in the low dimensional space. Two popular dimensionality reduction techniques, Principal Component Analysis (PCA) and t-distributed Stochastic Neighborhood Embedding (t-SNE) are described in Section 2.3 and Appendix A respectively.

After dimensionality reduction, clustering algorithms are used to classify data points corresponding to individual cells. Several clustering algorithms are publicly available and have been empirically evaluated using synthetic data for their performance, robustness and accuracy [29]. Although other algorithms may yield better results, the k-means algorithm is widely used for its ease of implementation and the availability of several parameter estimation techniques to estimate the parameter k, corresponding to the number of clusters in the data set.

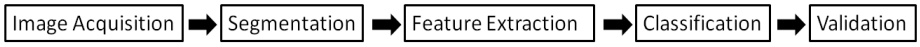


Figure 2.1: Standard pipeline for cell classification using microscopy data

Figure 2.1 illustrates the overall methodology organized in a data processing pipeline. The pipeline is implemented in a modular manner and individual components can be replaced if required. For instance, to classify fluorescent images instead of phase-contrast images, replacing the image segmentation method would suffice. Similarly, if the k-means classification algorithm is unable to find clusters in the input data or the algorithm fails to converge, then it can be easily swapped for a more sophisticated algorithm.

2.1 Image Processing

Identifying cells in an image is essential for automating the recognition of multiple cell types in large cell populations. Automated processing of 2-D images to count cells and identify cell types using morphological measurements has been steadily gaining traction since the 1960s. Over the past decades, literature on the subject has grown exponentially, with more than half of the bulk of papers appearing after the year 2000 [22].

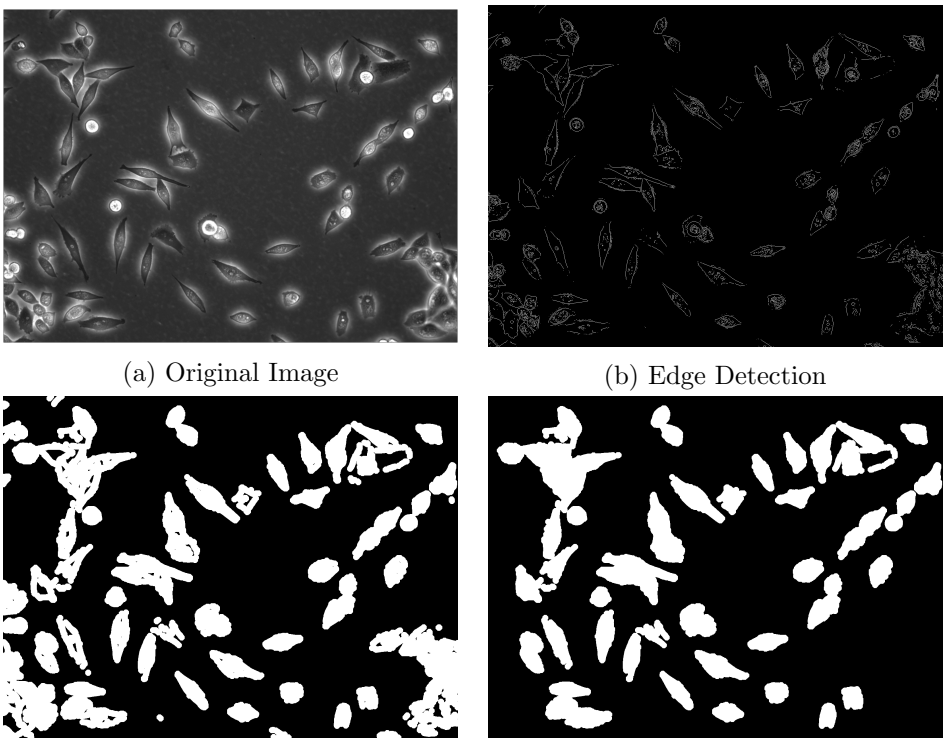
With the exception of neuron segmentation, the vast majority of current segmentation methods are based on few basic approaches; namely, intensity thresholding, feature detection, morphological filtering, region accumulation and deformable model fitting. These methods are reviewed in [22]. Region accumulation methods such as Voronoi-based methods or watershed transform can result in inaccurate cell boundaries by mis-specifications of the cell region to be divided or by over-segmentation. Similarly, popular deformable model approaches such as geodesic active contours or level sets, which detect cell boundaries by minimizing a predefined energy functional, can result in poor boundary detection because they use local optimization algorithms that only guarantee to find a local minimum or use the gradient vector field of the image to decode the boundary information [10].

Segmentation of bright field and phase contrast images is generally more challenging compared to fluorescent images. The latter usually have better contrast and deformable model fitting techniques like active contour or level sets work well [39]. Distinctive bright white patches or halo surrounding cells in bright field and phase contrast images prevent accurate determination of cell boundary. Therefore, a custom approach is required for each application that takes heterogeneity in cell shape, population density, variability in cell compartmentalization, etc., into account.

The following sections describe an approach for segmenting phase contrast images using a combination of edge detection, thresholding, mathematical morphology and watershed transform.

2.1.1 Foreground Detection

Foreground detection is performed in three stages. Firstly, the Sobel-Feldman derivative filter is applied to the original grayscale image to find edge points. These points are pixel locations in the image corresponding to non-zero intensity changes. The Sobel-Feldman operator uses two 3×3 kernels, one for derivative in a horizontal direction and the other for derivative in a vertical direction, which are convolved with the original image to calculate gradient approximation. The result is binarized by thresholding, with the value of the threshold specified by the user.



(a) Original Image (b) Edge Detection

(c) Dilated & Filled (d) Foreground Binary Mask

Figure 2.2: Foreground detection from phase contrast image
(a) Original phase contrast microscopy image of MIA PaCa-2 pancreatic carcinoma cell line courtesy of the Roskelley Lab at UBC. (b) Edge point detection by applying Sobel-Feldman derivative filter and conversion from grayscale to binary by thresholding. (c) Dilation by line-shaped structural element. (d) Resulting foreground markers obtained after filling, removal of small artifacts and objects connected with image boundary.

The binary image produced by edge detection is further manipulated using mathematical morphology, as shown in Figure 2.2. Mathematical morphology is a collection of set-theoretic operations on binary images that have been used for image enhancement, noise removal, edge detection, etc. Foundations of mathematical morphology are based on two operations: erosion and dilation.

Erosion of image A by structural element (abbreviated "strel") B, resulting

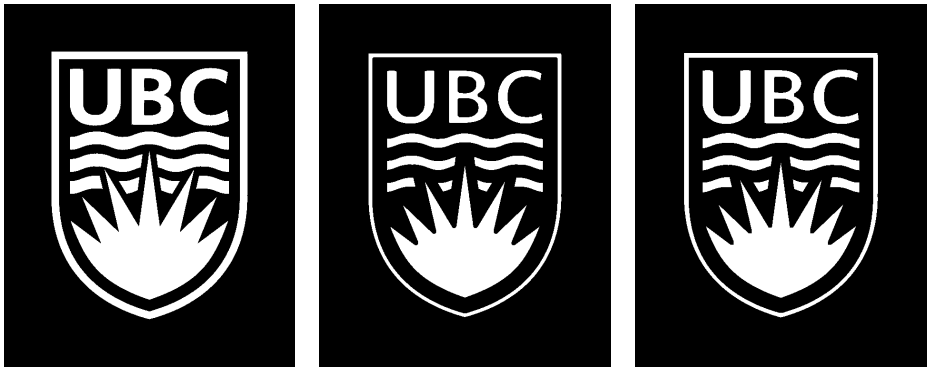
in image E is defined as:

$$A \ominus B = \{z \in E | B_z \subseteq A\},$$

where B_z is the translation of B by the vector z:

$$B_z = \{b + z | b \in B\}, \forall z \in E.$$

In practice, erosion leads to shrinking or thinning of the binary image, as shown in Figure 2.3.



(a) Original Image (b) Eroded by disk strel (c) Eroded by square strel

Figure 2.3: Mathematical morphology: image erosion
(a) Original image of UBC logo. (b) Image eroded by disk structural element with radius of 5 pixels. (c) Image eroded by square structural element with side length of 7 pixels. The diagonal length of the square element is 10 pixels, which is equivalent to the diameter of the disk element.

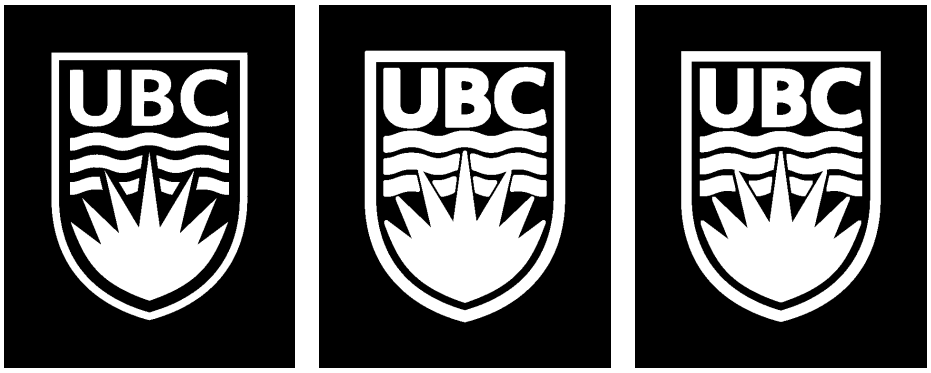
Dilation of image A by structural element B, resulting in image E is defined as:

$$A \oplus B = \bigcup_{z \in B} A_z,$$

where A_z is the translation of A by the vector z:

$$A_z = \{a + z | a \in A\}.$$

Dilation is used to grow or thicken regions in a binary image, as shown in Figure 2.4. All other morphological operations can be defined by composing erosions and dilations.



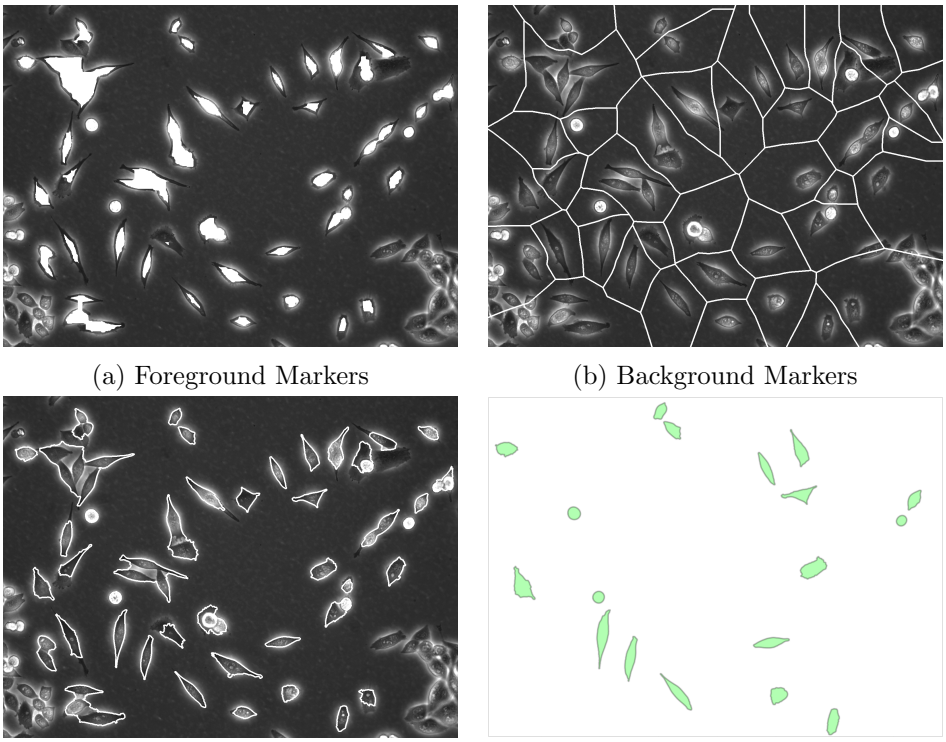
(a) Original Image (b) Dilated by disk strel (c) Dilated by square strel

Figure 2.4: Mathematical morphology: image dilation
(a) Original image of UBC logo. (b) Image dilated by disk structural element with radius of 5 pixels. (c) Image dilated by square structural element with side length of 7 pixels. Notice that corners are rounded in Figure 2.4b as a consequence of the shape of the disk, whereas corners remain sharp in Figure 2.4c.

In the second stage of foreground detection, edge points are connected by dilating the image with line shaped structural elements. The size of the structural elements is specified by the user. This leads to the formation of closed loops around isolated cells or clusters of tightly packed cells. The final stage involves filling of closed loops (using imfill in MATLAB), removal of small objects whose size is below user-specified threshold and removal of objects that cross the image boundary.

Cell Segmentation

Foreground detection resulted in separation of foreground and background from the original image. The foreground binary image is eroded multiple times (number of erosions and size of structural element is specified by the user) to obtain foreground markers, a majority of which lie inside cell boundaries. The marker-based watershed transform is a region accumulation approach that segments cell boundaries using foreground markers and gradient of the original image. The number of correct segmentations in the result depends on the pre-processing of markers prior to the segmentation [36].



(a) Foreground Markers (b) Background Markers

(c) Watershed Segmentation (d) Manual Selection

Figure 2.5: Cell segmentation
(a) Eroded binary foreground markers overlaid on top of original image. (b) Background markers computed by applying watershed transform to the distance transform of foreground markers. (c) Result of watershed segmentation using foreground and background markers. (d) Manual selection of correctly segmented cells.

The watershed segmentation algorithm requires both foreground and background markers. Foreground markers specify regions inside individual cells whereas background markers specify regions between adjacent cells. To prevent over-segmentation due to background markers being too close to objects of interest (i.e. cells), background markers are computed by calculating the skeleton by influence zones (SKIZ) of the foreground markers [24]. The influence zone of a foreground marker is the set of neighboring pixels that are closer to that foreground marker than to any other foreground markers. SKIZ is the boundary between influence zones of all foreground markers. It is analogous to the Voronoi tessellation of foreground markers in the image

plane. In practice, the background markers are computed using a simple two-step procedure. Firstly, the distance transform of the foreground markers is computed. Then, ridge lines (corresponding to background markers or SKIZ, as shown in Figure 2.5b) are determined by computing the watershed transform of the distance-transformed foreground markers [32]. Once both foreground and background markers are computed, the priority-flood watershed algorithm is applied to the original image, resulting in watershed lines corresponding to the boundary of the cells as shown in Figure 2.5c. An outline of the watershed algorithm follows:

Priority-Flood Watershed Algorithm [3]:
Step 1: Foreground and background markers are chosen. Each set of connected markers is assigned a different label.
Step 2: The neighboring pixels of each marked area are inserted into a priority queue (a list of objects sorted by their priority level), with a priority level corresponding to the magnitude of the gradient of intensity at that pixel.
Step 3: The pixel with the lowest priority level is extracted from the priority queue. If all labeled neighbors of the extracted pixel have the same label, then the pixel is assigned their label. All non-marked neighbors that are not yet in the priority queue are put into the priority queue.
Step 4: Step 3 is repeated until the priority queue is empty.

The entire image segmentation process requires minimal user input. The user has to specify the threshold parameters, size of structural elements and number of iterations for morphological operations. Finally, the user is required to manually select correct segmentation from the result of applying the watershed transform. This ensures that only correctly segmented cells are assigned unique ID numbers (and serialized) for further processing in the pipeline.

2.2 Feature Extraction

In general, there are two approaches for extracting morphological features from segmented cell shapes: boundary-based methods that extract information from points on the cell boundary and area-based methods which use all points on the interior and boundary of the cell shape. The area based methods are more robust to small perturbations in cell shape and are easy to compute. For example, to accurately estimate the area of a given shape it

is sufficient to count the number of pixels that make up the shape, whereas perimeter estimation is not so straightforward [18]. The main advantage of boundary based features such as curvature functions, cubic spline interpolation of cell boundary, normalized Fourier shape descriptors, etc., is that they provide a good quantization of angles, corners and curves in the image. These details are lost when summing over all image pixels to compute area based features like Hu's moment invariants and non-dimensional shape factors.

2.2.1 Hu's Moment Invariants

The mathematical concept of moments can be used to quantify global properties of an image. Global image properties refer to an image as a whole rather than its components. Consider a binary segmented cell image defined by intensity function $f : \mathbb{R}^2 \to \{0, 1\}$ that maps each pixel (x, y) to 0 or 1 depending on whether the pixel lies outside or inside/on the cell boundary. Raw image moments m_{pq} of order $p + q$ are defined as projections of image $f(x, y)$ to basis $x^p y^q$:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy.$$

In a discrete setting, the integral is replaced by a sum over all pixels in the raster image:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y).$$

The zeroth raw moment, m_{00}, is the sum of intensities over all pixels in the image. For a binary image, where pixel intensity is equal to 1 if the pixel is part of the cell and 0 otherwise, m_{00} corresponds to the area of the cell. $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$ are coordinates (x, y) of the centroid. To confer translational invariance, one can compute central image moments about the mean:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y).$$

It can be easily verified that the zeroth central moment, μ_{00}, is equivalent to m_{00} and it corresponds to the area of the segmented cell.
Now, consider an image scaled by factor λ: $f'(x, y) = f(x/\lambda, y/\lambda)$. The scaled image (defined by function f') can be smaller (if $\lambda > 1$) or larger

(if $\lambda < 1$) compared to the original image (defined by function f). Central moments of the scaled image are given by:

$$\mu'_{pq} = \int\int x^p y^q f(x/\lambda, y/\lambda) dx dy.$$

Substituting $x' = x/\lambda$ and $y' = y/\lambda$, we derive:

$$\mu'_{pq} = \int\int (\lambda x')^p (\lambda y')^q f(x', y') \lambda^2 dx' dy',$$

$$\mu'_{pq} = \lambda^{(p+q+2)} \mu_{pq}.$$

Therefore, normalized central moments that are translation and scale invariant can be obtained by setting area (μ'_{00}) equal to unity:

$$\mu'_{00} = \lambda^2 \mu_{00} \implies \lambda = \mu_{00}^{-\frac{1}{2}}.$$

Normalized central moments of order $p + q$ are typically denoted by η:

$$\eta_{pq} = \mu_{00}^{\frac{-(p+q+2)}{2}} \mu_{pq}.$$

However, a similar calculation corresponding to image rotated by angle θ: $f'(x, y) = f(x\cos(\theta) + y\sin(\theta), -x\sin(\theta) + y\cos(\theta))$ does not yield rotationally invariant moments. Rotational invariance requires a nonlinear transformation that is not trivial to compute. Hu derived these nonlinear expressions from normalized central moments up to order three using algebraic invariants [16]. Hu's moment invariants are widely used for translation, scaling and rotation invariant pattern recognition, including recognition of typed English language characters [17]. Typically a feature vector for image classification is comprised of seven invariants:

$$\phi_1 = \eta_{20} + \eta_{02},$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2,$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2,$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],\end{aligned}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].\end{aligned}$$

Note that ϕ_7 has the additional property of being skew invariant and therefore can be used to distinguish between mirror images. Unlike raw or central moments, these finite order invariant moments do not form a complete set of image descriptors. While higher order moments can be calculated, image reconstruction given a set of Hu's moment invariants is not straightforward. Furthermore, all seven invariant moments are zero for images that are rotationally symmetric [30].

Dunn and Brown used shape measures (extension, dispersion and elongation) and principal axis orientation calculated using ϕ_1 and ϕ_2 to characterize the shape and alignment adopted by chick heart fibroblasts on microfabricated grooved substrata [11]. However, recent literature on morphology-based cell classification omits the use of Hu's moment invariants. The role of these invariants and their usefulness is investigated in Chapter 3 and Appendix A.

2.2.2 Geometrical and Boundary Features

In order to achieve high-throughput cell classification, the feature vector describing the shape of the cell should be concise and computationally inexpensive to calculate. Hu's moment invariants meet this criterion but do not have any intuitive meaning. Given an arbitrary set of Hu's moment invariants, one cannot easily imagine the shape of the object that produced those moments.

Normalized Fourier shape descriptors (FSD) represent the boundary of an object using a subset of coefficients from the Fourier transform of its contour [13]. However, the number of coefficients required to accurately represent a given object depends on the curvature of the object and usually a large number is required. Furthermore, it is not evident how the number of coefficients required can be estimated *a priori* without trial and error.

Curvature functions are computed from the contour of an object and describe how much a curve bends at each point. Peaks in the curvature function correspond to corners on the object. Urdiales et al. describe a non-parametric method for efficient computation of a short feature vector for a planar shape using its curvature function [40]. However, in order to produce a short feature vector, their algorithm requires pre-computing a set of representative curvature functions by calculating the Fourier transform of curvature functions of typical shapes. Then, for a given input shape, the

algorithm returns similarity measures of the shape's curvature function compared to the representative set. The pre-computation step is not desirable for high-throughput cell classification (particularly for on-line classification where the entire data set is not available in advance), therefore curvature function-based features are omitted.

In view of the above, geometrical descriptors included in the feature set constitute estimates of dimensions of the cell obtained by fitting conics (circles and ellipses) as well as uncertainty in those estimates obtained from goodness of fit measures. Conic fitting is universally applicable to all planar objects and requires only a set of boundary points (obtained from segmentation) as input. Boundary descriptors are obtained from cubic spline interpolation of the boundary of the cell, where the number of spline points is estimated using a manually adjusted smoothing parameter. These descriptors encode information about the shape and size of the cell that is easy to visualize (as shown in Figure 2.6) and understand. Like Hu's moment invariants, these features are resistant to affine transformations (scaling, rotation and translation) as well as to noise in the shape boundary.

Consider an arbitrary geometry $f(\theta) = 0$ parametrized by M features, $\theta = (\theta_1, ..., \theta_M)^T$. To fit this geometry to a set of boundary points $(x_i, y_i)_{i=1}^N$, consider the following optimization problem:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} r_i^2(\theta),$$

where r_i is the orthogonal distance between boundary point (x_i, y_i) and shape $f(\theta) = 0$.

For example, to fit a circle $f(\theta) = 0 \iff x^2 + y^2 - r^2 = 0$, where $\theta = (r)$. Similarly, for ellipse fitting, $f(\theta) = 0 \iff ax^2 + by^2 + cxy + dx + ey + f = 0$, where $\theta = (a, b, c, d, e, f)$. However, the application of ordinary least squares does not return consistent estimates for parameter(s) θ. Therefore, typically one of the following techniques are used: gradient-weighted least squares fitting, bias-corrected renormalization fitting or extended Kalman filtering [6].

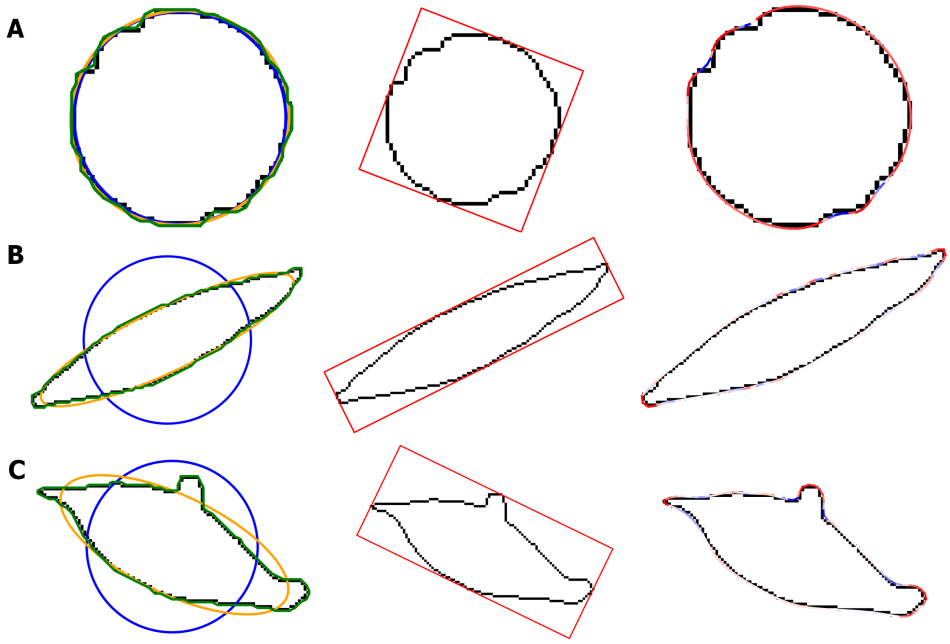


Figure 2.6: Geometrical fits and boundary interpolation
Left: Circle, ellipse and polygon fits for three distinct cell geometries. Middle: Rectangle fit is used to compute maximum and minimum Feret diameter. Right: Cubic spline interpolation of cell boundary colored by magnitude of curvature.

Ellipse and Circle Fitting

Consider an ellipse centered at (x_c, y_c) and rotated by angle α. Let (x_t, y_t) be the closest point on the ellipse to boundary point (x_i, y_i). Then the shortest distance D_i from the boundary point to the ellipse is given by:

$$x_t = x_c + a\cos(\alpha)\cos(t) - b\sin(\alpha)\sin(t),$$

$$y_t = y_c + a\sin(\alpha)\cos(t) + b\cos(\alpha)\sin(t),$$

$$D_i = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2}.$$

In this case, the optimal least squares solution can be computed directly without an iterative approach [15]. The stable and robust fitting method returns parameters $\theta = (x_c, y_c, a, b, \alpha)$. In addition to parameters obtained from fitting, goodness of fit is estimated by calculating its variance as follows [28]:

Suppose $(\bar{x}, \bar{y})$ is the centroid and $(x_i, y_i)_{i=1}^{N}$ are the boundary points on the contour of the shape that is being fitted. Then covariance matrix of the contour is:

$$C = \frac{1}{N}\sum_{i=1}^{N} V_i V_i^T = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix},$$

where

$$V_i = \begin{pmatrix} x_i - \bar{x} \\ y_i - \bar{y} \end{pmatrix},$$

and,

$$c_{xx} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2,$$
$$c_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}),$$
$$c_{yx} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})(x_i - \bar{x}),$$
$$c_{yy} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2.$$

Lengths of the two principal axes of the ellipse fit can be obtained by calculating eigenvalues of the covariance matrix:

$$det(C - \lambda_{1,2} I) = 0,$$
$$\lambda_1 = \frac{1}{2}[c_{xx} + c_{yy} + \sqrt{(c_{xx} + c_{yy})^2 - 4(c_{xx}c_{yy} - c_{xy}^2)}],$$
$$\lambda_2 = \frac{1}{2}[c_{xx} + c_{yy} - \sqrt{(c_{xx} + c_{yy})^2 - 4(c_{xx}c_{yy} - c_{xy}^2)}],$$
$$\text{Ellipse eccentricity, } e = \frac{\lambda_2}{\lambda_1}.$$

Variance is the standard deviation of radial distance from the centroid to the boundary points divided by the mean. Variance close to zero indicates a good fit.

$$\text{Variance of fit} = \frac{\sigma_R}{\mu_R},$$

where,

$$\mu_R = \frac{1}{N}\sum_{i=1}^{N} d_i,$$

$$\sigma_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \mu_R)^2},$$

$$\text{and} \quad d_i = \sqrt{V_i^T C^{-1} V_i}.$$

The following table summarizes features obtained from ellipse and circle fitting:

Feature	**Range**	**Description**
Ellipse Eccentricity	$[0, 1]$	Close to 0, the ellipse is circular. Close to 1, it is elongated.
Ellipse Major Axis Length	$[0, \infty)$	The length of the major axis of the ellipse fit.
Ellipse Minor Axis Length	$[0, \infty)$	The length of the minor axis of the ellipse fit.
Ellipse Area	$[0, \infty)$	Area of the ellipse fit.
Ellipse Perimeter	$[0, \infty)$	Perimeter of the ellipse fit.
Ellipse Variance	$[0, 1]$	A goodness of fit measure for the ellipse fit.
Circle Radius	$[0, \infty)$	Radius of the circle fit.
Circle Area	$[0, \infty)$	Area of the circle fit.
Circle Variance	$[0, 1]$	A goodness of fit measure for the circle fit.

Table 2.1: Features extracted from ellipse and circle fits of cell images

Rectangle Fitting

Fitting a bounding rectangle to boundary points obtained from segmentation is different from fitting conic sections because a rectangle cannot be described by a single continuous function. A rectangle consists of four such functions with constraints between them. Chaudhuri and Samal describe a step-wise procedure: 1) finding the centroid of the object, 2) determining principal axes, 3) computing the upper and lower furthest edge points along

the boundary, and finally, 4) finding the vertices of the bounding rectangle [8]. The following table summarizes features obtained from rectangle fit of a cell. These features are used in computation of elongation (a non dimensional shape factor) below. MoHan Zhang, an undergraduate student at UBC, implemented this method as part of a research experience program project.

Feature	**Description**
Maximum Feret Diameter	Furthest distance between any two parallel tangents on the cell
Minimum Feret Diameter	Shortest distance between any two parallel tangents on the cell

Table 2.2: Features extracted from bounding rectangle fits

Polygon Fitting

A polygon fit along the cell boundary is computed using the 3-pixel vector (3PV) method described by Inoue and Kimura [18]. The 3PV method is designed for calculating the perimeter of low resolution raster objects, where counting the number of pixels at the boundary of the object results in inaccuracies. Starting from an arbitrary location, adjacent boundary pixels are assembled in an ordered set. Each element in the set is an ensemble of three adjacent pixels enumerated in counterclockwise order. The spatial configuration of each 3-pixel ensemble is specified using a pair of integers from 0 to 7. The integers specify the direction of counterclockwise travel between consecutive pixels. Integers 0, 2, 4, 6, represent east, south, west and north directions respectively, as shown in Figure 2.7. Similarly, integers 1, 3, 5, 7, are used to encode southeast, southwest, northwest and northeast directions respectively for diagonal placement of pixels. Therefore, a 3-pixel ensemble corresponding to "L" shape (Figure 2.7) is represented by integers $[4, 6]$ indicating west and north direction of travel in counterclockwise manner. The ordered set of integer representations for 3-pixel ensembles starting from the arbitrary location is defined as the chain code of the object boundary.

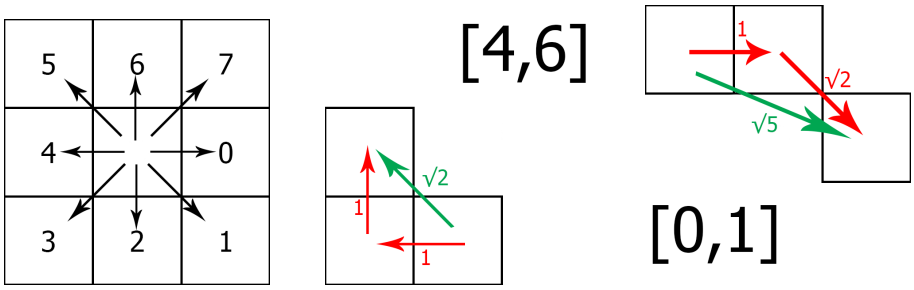


Figure 2.7: Computing chain code from boundary pixels
(a) Reference diagram from computing chain code. (b) An example of 3-pixel configuration corresponding to $\sqrt{2}$ length. (c) An example of 3-pixel configuration corresponding to $\sqrt{5}$ length.

As part of geometrical feature extraction, 3PV method is used to compute the perimeter of segmented cell images from the chain code representation of the cell boundary. Adjacent pair of elements in the chain code is referred to as a chain pair. Inoue and Kimura [18] specify corrections to typical perimeter calculation (the so-called $1, \sqrt{2}, \sqrt{5}$ method, illustrated in Figure 2.7, for computing distances for straight, diagonal, and straight followed by diagonal placement of pixels respectively) for all possible combinations of chain pairs. A set of vectors consisting of pairs of points along the cell boundary (called 3-pixel vectors, since they are derived from chain pairs corresponding to 3-pixel ensembles) is computed where the sum of lengths of these vectors provides an accurate estimate of the perimeter of the cell boundary. With minor adjustments (to account for cases where adjacent vectors do not align head to tail) as shown in Figure 2.8, the 3-pixel vector is used to obtain a polygonal fit to the cell geometry. Darrick Lee, an undergraduate summer research student at UBC, implemented the 3PV method and the cubic spline interpolation of cell boundary (derived from the polygonal fit) described in the next section.

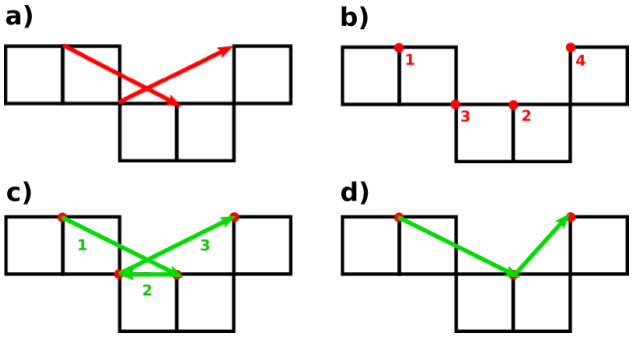


Figure 2.8: Computation of polygonal fit from chain code

(a) 3-pixel vectors obtained from chain code $[0, 1, 7]$ corresponding to east, southeast and northeast direction of travel between two adjacent 3-pixel ensembles on the cell boundary. (b) Ordered set of points along the cell boundary corresponding to the 3-pixel vectors. (c) Connecting points in order results in incorrect polygon segment as 3-pixel vectors are not aligned head to tail. (d) A vertex is removed to correct the boundary segment.

Feature	**Method**	**Description**
Cell Area	Pixel Counting	Number of pixels inside segmented cell boundary.
Cell Perimeter	3-pixel Vector (3PV) Method	Estimate obtained from chain code for pixels on cell boundary.

Table 2.3: List of other geometrical features

Cubic Spline Boundary Fitting

A cubic spline interpolation along the cell boundary is computed from vertices of the polygon fit described in the previous section. Curvature is calculated by sampling points on the spline. As expected, the number of changes in sign (positive to negative and vice-versa) correlates with convexity of the cell shape. Convex circular cells have positive curvature throughout the boundary and zero changes in curvature sign. For cells with non-zero number of sign flips, areas of high positive curvature correspond to protrusive regions on the boundary. Boundary descriptors in the feature vector include number of changes in the curvature sign (i.e. number of zero crossings), global maximum and global minimum of curvature.

Feature	Description
Number of sign flips	Number of times curvature changes sign from positive to negative and vice-versa.
Maximum curvature	Absolute maxima of curvature on cell boundary.
Minimum curvature	Absolute minima of curvature on cell boundary.

Table 2.4: Features extracted from curvature of cubic spline fit

2.2.3 Shape Factors

The previous section described features obtained by fitting various geometries to a cell boundary. Shape factors are non-dimensional quantities that are computed by counting pixels in a segmented cell image, and its convex hull, bounding box and bounding rectangular fit. Note the distinction between bounding box and the rectangle fit. The edges of the bounding box are parallel to Cartesian axes whereas the major axis of the bounding rectangular fit is aligned to the principal axis of the cell shape. Shape factors are widely used to classify particulate matter [5, 27] and often used as part of feature vectors designed to classify cell shapes [7, 31].

Extent

The ratio of the number of pixels belonging to a segmented cell to the number of pixels in its bounding box is defined as the extent. The bounding box spans horizontally from the leftmost pixel to the rightmost pixel and vertically from the topmost pixel to the bottommost pixel. Extent is close to zero if a cell is elongated and close to unity if the cell is uniformly spread out.

Solidity

Solidity is a measurement of the overall concavity of an object. It is defined as the ratio of the number of pixels belonging to the segmented cell to the number of pixels in its convex hull. As cell shape deforms from a convex polygon or circle to a more elliptical or protrusive shape, its convex hull area increases compared to the cell area and solidity correspondingly decreases.

For the MIA PaCa-2 pancreatic cancer data set, rounded cells typically have solidity values that approach unity.

Compactness

Compactness is defined as the ratio of the circular equivalent diameter to the maximum Feret diameter obtained from the bounding rectangular fit:

$$\text{Compactness} = \frac{\sqrt{\frac{4(A_{\text{cell}})}{\pi}}}{\text{Max. Feret diameter}}$$

The circular equivalent diameter, also known as area-equivalent diameter, is defined as the diameter of a circle with the same area as the object. Like extent, compactness is close to zero if a cell is elongated or 'I' shaped.

Elongation

Elongation is defined as (1 − Aspect Ratio). Aspect ratio is obtained from the rectangle fit as the ratio of minimum to maximum Feret diameters. For elongated cells, maximum Feret diameter is much larger than minimum Feret diameter, therefore their elongation is close to unity. Conversely, for circular cells, both diameters are roughly the same. Therefore the elongation of such cells is close to zero.

Circularity

Circularity measures the degree to which an object is similar to a circle:

$$\text{Circularity} = \sqrt{\frac{4\pi A_{\text{cell}}}{P_{\text{cell}}^2}}$$

It can be easily verified that circularity for a perfect circle is unity. Regular polygons approach a circle as their number of edges increases. It should be noted that a low value of circularity does not necessarily mean that the cell shape lacks rotational symmetry. Circularity close to zero typically indicates elongated or protrusive (e.g. starfish-like) morphology.

Convexity

Convexity is defined as the ratio of the convex hull perimeter and the actual perimeter of the object. It is highly sensitive to deviations from convex geometry. Convexity is close to zero for highly non-convex cell geometries

and close to unity for epithelial-like cells with polygonal morphology (absent from MIA PaCa-2 data set) or circular cells.

The following table describes non-dimensional shape factors included in the feature vector and formulas for their computation:

Feature	**Range**	**Equation**	**Description**
Extent	$[0, 1]$	$\frac{A_{\text{cell}}}{A_{\text{bounding box}}}$	Ratio of pixels belonging to segmented cell to pixels in the bounding box.
Solidity	$[0, 1]$	$\frac{A_{\text{cell}}}{A_{\text{convex}}}$	Ratio of pixels belonging to segmented cell to pixels in the convex hull.
Compactness	$[0, 1]$	$\frac{\sqrt{\frac{4(A_{\text{cell}})}{\pi}}}{\text{Max. Diameter}}$	Ratio of circular equivalent diameter to maximum Feret diameter.
Elongation	$[0, 1]$	$1 - \frac{\text{Min. Diameter}}{\text{Max. Diameter}}$	1 - Aspect Ratio. Close to 1 for elongated cells and close to 0 for circular cells.
Circularity	$[0, 1]$	$\sqrt{\frac{4\pi A_{\text{cell}}}{P_{\text{cell}}^2}}$	Degree of resemblance to a circle.
Convexity	$[0, 1]$	$\frac{P_{\text{convex hull}}}{P_{\text{cell}}}$	Ratio of the convex hull perimeter to the cell perimeter.

Table 2.5: List of non-dimensional shape factors

2.3 Dimensionality Reduction

High-dimensional data exhibits "curse of dimensionality", i.e. distances between all pairs of points converges to the same value in higher dimensions. Therefore, unsupervised classification methods that rely on clustering algorithms to categorize the data set using some distance metric fail to perform well for high dimensional feature vectors. To prevent this problem, clustering is typically performed on a low-dimensional data set after dimensionality reduction.

2.3.1 Principal Component Analysis (PCA)

Principal Component Analysis is a mathematical technique that exploits variance in a given data set in order to make patterns in the data salient. It is commonly used in machine learning to visualize and manipulate high dimensional feature vectors. PCA transforms high dimensional data into a low dimensional subspace, where the basis for the low dimensional subspace is a linear combination of high dimensional basis vectors. The linearity assumption reduces the problem of finding an appropriate transformation to the problem of finding an appropriate projection. The linear combination of the original basis is determined in a manner such that the low dimensional basis vectors (also known as principal components) correspond to direction with the greatest variance in the data. In other words, PCA projects high dimensional feature vectors to a new (low dimensional) coordinate system, ensuring that the first axis in the new coordinate system (PCA 1) has maximum variation, the second axis (PCA 2) has the second-most variation, and so on.

Mathematically, the principal components are the eigenvectors of the covariance matrix of the original data set. These eigenvectors are orthogonal since the covariance matrix is symmetric. Reducing dimensionality of the feature space by projecting all feature vectors to a low dimensional space reduces the complexity of cluster identification and k-means clustering. The number of principal components (i.e. dimensionality of the transformed space) is chosen by plotting the variance in data explained by each principal component versus the number of components. Typically, the number of principal components is determined by finding an "elbow" in this plot. The elbow signifies the turning point where the trade-off between including additional variance is offset by the complexity of dealing with more components. Generally, most of the variance in the original data is explained by a small number of principal components. Sometimes 2 or 3 components are chosen for ease of plotting. As a rule of thumb, in this thesis, the number is components is chosen such that the total explained variance exceeds 80%.

2.4 Unsupervised Classification

Unsupervised classification refers to grouping of quantifiable objects by inferring relationships between these objects. Clustering algorithms are used to automatically group the data points (also called descriptors or features

in the context of machine learning) corresponding to the objects of interest. The terms "clustering" and "unsupervised classification" are used interchangeably throughout this thesis. Multiple methods for clustering exist and they are generally categorized into one of four types [37]:

Prototype-Based Methods: Objects belong to the same cluster if their distance (a measure of similarity) to the prototype that defines the cluster is smaller compared to their distance to prototypes of other clusters. The prototype is the most representative point of a given cluster, often lying at the center of that cluster. k-means, k-medoids and X-means algorithms belong to this category.

Graph-Based Methods: Relationships between objects (nodes in this context) are represented by edges between nodes in a graph. Objects are clustered by identifying connected components, cliques and neighboring nodes in the graph. Some advanced algorithms define criteria for splitting edges in the Minimum Spanning Tree (MST) of the graph to obtain a forest of clustered points. The Fuzzy C-Means MST Clustering algorithm, Markov Clustering algorithm, Iterative Conductance Cutting algorithm, Geometric MST Clustering algorithm and Normalized Cut Clustering (NCC) algorithm belong to this category [12].

Density-Based Methods: Objects are represented by points in space. A cluster refers to group of densely packed points surrounded by a region of low density. DBSCAN (density-based spatial clustering of applications with noise) and OPTICS (ordering points to identify the clustering structure) algorithms are examples of density-based approach to clustering.

Shared Property or Conceptual Methods: This is a catch-all category that defines a cluster as groups of objects that share some common property.

Irrespective of the method used for clustering the data, it is important to find parameters to optimize algorithm performance and ensure that the results obtained from the algorithm are meaningful. Many clustering algorithms are known to be very sensitive to their input parameters [19]. Measures for assessing the efficacy of clustering like Davies-Bouldin index and silhouette score (defined below) are useful for evaluating the results of clustering algorithms.

The k-means algorithm and DBSCAN are two widely used methods for clustering data in low dimensions. While the k-means algorithm requires one parameter (k), the DBSCAN algorithm requires two parameters (minPts, the minimum number of points in the neighborhood of a core point and neighborhood distance ϵ) for distinguishing between core points, boundary points and non-reachable points. The k-means algorithm is used for clustering data in this thesis due to its ease of implementation and availability of methods for parameter estimation. Due to the modular nature of the unsupervised cell classification pipeline, the k-means clustering algorithm can be swapped for more sophisticated algorithms like X-means or OPTICS that do not require any input parameters in the future.

2.4.1 K-means Clustering Algorithm

The k-means algorithm partitions input data points, $\mathbf{x_i}$ $(i = 1, \ldots, N)$, into k subsets or clusters, where points in a cluster C_p $(p \in \{2, \ldots, k\})$ are associated with cluster center $\mu_\mathbf{p}$. The partitioning is determined by minimizing the distance between points and their cluster centers according to some user-specified distance metric.

K-means Algorithm:
Step 1: Randomly pick k points $\mu_\mathbf{1}, \ldots, \mu_\mathbf{k}$, from input data $\mathbf{x_i}$ as cluster centers. Initialize k clusters, $C_1, \ldots, C_k$, with these points.
Step 2: Assign each point $\mathbf{x_i}$ to the "nearest" cluster, C_p. The nearest cluster is determined by minimizing the sum of squared intra-cluster distances between points and their cluster centers:

$$\underset{p}{\operatorname{argmin}} \sum_{\mathbf{x_j} \in C_p} (\mathrm{D}(\mathbf{x_j}, \mu_\mathbf{p}))^2.$$

Typically, Euclidean measure is used to compute distance D.
Step 3: Calculate new cluster centers by computing the mean of points in each cluster:

$$\mu_p = \frac{1}{|C_p|} \sum_{\mathbf{x_j} \in C_p} \mathbf{x_j}$$

Step 4: Repeat steps 2 and 3 until cluster centers stop changing.

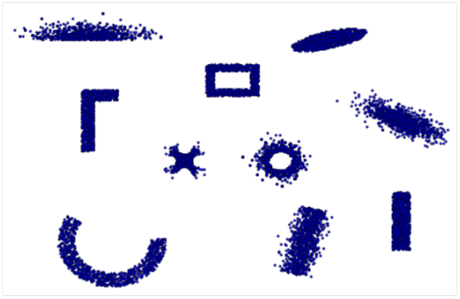
Typically, the algorithm is run multiple times with different random initial choices of cluster centers to ensure stable convergence. In addition to the input data points, the user has to supply parameter, k, which denotes the

number of clusters in the data. There are four popular methods to find the optimal number of clusters in an arbitrary data set: elbow heuristic, Bayesian information criterion (abbreviated BIC), silhouette analysis [34] and gap statistic [38]. In practice, using features computed from the MIA PaCa-2 data set, silhouette analysis (described below) performed best in terms of robustness and convergence.

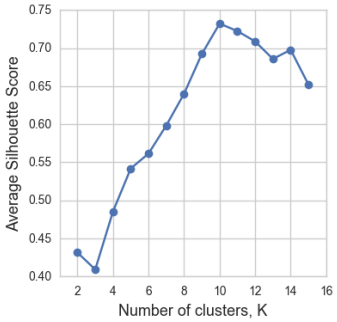
2.4.2 Silhouette Score Analysis

Silhouette analysis is the study of the degree of separation between clusters of data points using silhouette coefficients. The silhouette coefficient for a given data point, P, is a measure that quantifies the degree to which the data point belongs to its assigned cluster, C. It is computed as follows. Let a be the mean distance between point P and every other point in its own cluster C. Let b be the mean distance between P and every point in the nearest neighboring cluster. Then, the silhouette coefficient for point P is $(b - a)/\max(a, b)$. The silhouette coefficient ranges from -1 to 1. A coefficient value near 1 indicates that P has undoubtedly been classified correctly, a value around 0 indicates that the clustering of P has some ambiguity, and a value near -1 indicates it is likely that P was classified incorrectly.

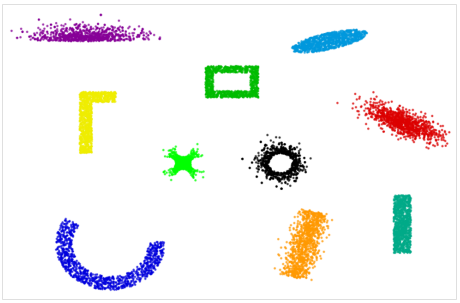
Rousseeuw described a heuristic using silhouette coefficients to identify the number of clusters in a given data set [34]. Points are clustered using k-means for various values of parameter k. Assuming that the algorithm converges and gives stable results, the silhouette score is computed by calculating the average of silhouette coefficients for all data points. The number of clusters in the data set, i.e. the optimal value for k, is one that maximizes the silhouette score. This technique is demonstrated using synthetically generated data in Figure 2.9. 10,000 data points corresponding to 10 clusters (1000 points per cluster) are generated by transforming and combining uniform random distributions (see Figure 2.9a). Figure 2.9b shows a plot of silhouette score computed for various values of parameter k. The most probable value of k is automatically determined by finding the maximum in this plot. For the synthetic data set, the silhouette score is maximized at $k = 10$. Figure 2.9c shows the k-means clustering result (corresponding to $k = 10$), with points colored according to their cluster label. Sorted values of silhouette coefficients for individual data points (grouped by cluster label) are shown in Figure 2.9d, with the vertical red line depicting the overall silhouette score obtained by averaging.



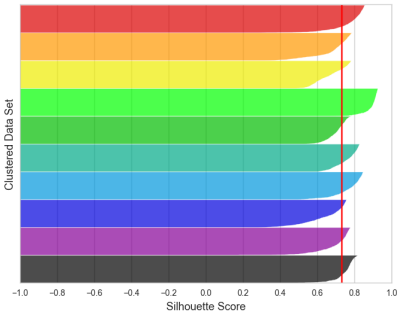
(a) 10,000 synthetic data points corresponding to 10 user-defined cluster shapes



(b) Determining optimal k automatically by computing average silhouette score



(c) Labeled data points (obtained by k-means) corresponding to 10 clusters



(d) Sorted silhouette scores for data points in each cluster

Figure 2.9: Clustering synthetic data using silhouette score analysis

Chapter 3

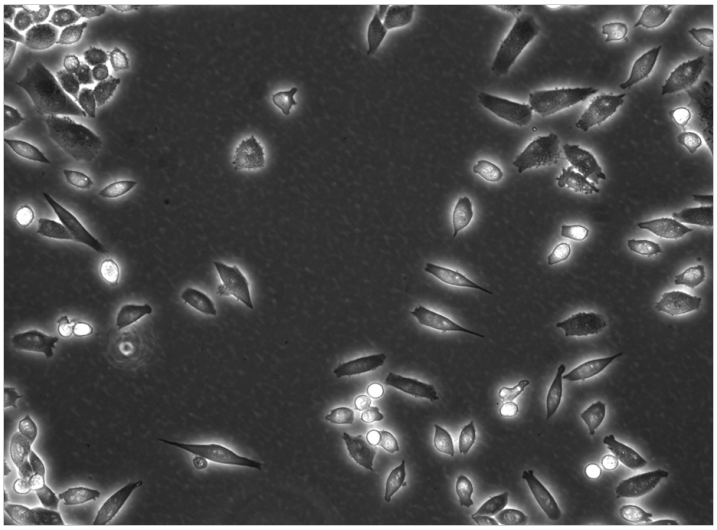
Results and Discussion

This chapter illustrates the methodology described in Chapter 2 using data acquired by segmenting phase-contrast microscopy images of the MIA PaCa-2 cancer cell line. MIA PaCa-2 is a human cell line that was established by A. Yunis, et al. [42] from primary tumor tissue of the pancreas. It is currently used as an *in-vitro* model to study carcinogenesis in pancreatic ductal adenocarcinoma [14]. The morphological and genetic characteristics of cells belonging to the MIA PaCa-2 cell line are well understood and readily available in the literature [14]. Therefore, unsupervised classification of these cells based on their morphology is of little biological relevance. This cell line was chosen for the purpose of demonstrating the methodology, since MIA PaCa-2 cells exhibit several distinct morphological shapes.

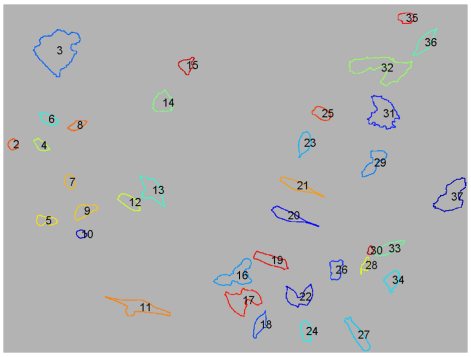
Phase-contrast MIA PaCa-2 images are segmented using the methodology described in Section 2.1. Using 40 images acquired by the Roskelley Lab at UBC, 149 correctly segmented cells are identified by visually inspecting the segmentation result and comparing it to the original phase image. A typical segmentation result is shown in Figure 3.1. Cells that cross the boundary of the image are removed during mathematical morphology stage of image processing. Note that boundaries of cells that are closely packed cannot be resolved by the watershed algorithm (see Figure 3.1b), since these cells share foreground markers and are treated as one object. The number of iterations required to separate foreground markers (obtained from mathematical morphology) of closely packed cells using erosion is high. Since all foreground markers are eroded equal number of times to keep the image segmentation process as automated as possible, too many erosion operations prevents correct segmentation of elongated cells due to the loss of foreground markers in their long thin "tails". Therefore, the number of iterations of erosion is a parameter that affects the range of cell sizes and population density at which correct segmentations can be obtained. Once this parameter is manually chosen, the user must identify correctly segmented cells (by selecting IDs assigned to watershed outlines, as shown in Figure 3.1b) for serialization. Serialization refers to the process of saving segmented boundaries in a

database for feature extraction and further processing.

Each cell is assigned a unique ID (henceforth referred to as UID) after all images are segmented and manually selected cells are serialized. UID is used to keep track of cells (and the original image from which they are obtained) through the remainder of the classification process.



(a) MIA PaCa-2 image acquired using phase-contrast microscopy



(b) Boundaries (tagged with IDs) obtained using mathematical morphology and watershed segmentation

(c) Correctly segmented cells are serialized by manually selecting IDs (from Figure 3.1b)

Figure 3.1: Selection of correctly segmented cells

As proof-of-concept, a manually curated subset of the 149 segmented cells is used for unsupervised classification in this thesis. The subset, consisting of 63 segmented cells, is preselected to only include cells with clear and distinct morphologies. The remaining cells with ambiguous morphology are omitted from the data set as follows. A preliminary set of features is computed for all 149 cells. Each feature vector is reduced to a two dimensional vector using principal component analysis (PCA). Outlier points (corresponding to anomalous cell morphologies) are identified and manually removed from the data set. UIDs of outliers are noted for further inspection as these cells are often of interest to experimental biologists. Manual outlier removal is a time-consuming task that does not scale with input data size. Automatic techniques for outlier detection (currently under development) will remove this bottleneck in the future. In addition, points that appear to lie between clusters in the two-component PCA space are removed to retain groups of cells that have distinctive morphologies. Removal of points between clusters improves stability of the k-means clustering algorithm. Stability and convergence of the algorithm is crucial for estimating parameter k (corresponding to number of clusters in the data set) using silhouette score analysis.

Four distinct morphologies are evident in the MIA PaCa-2 phase images (see Figure 3.2). The manually curated data set contains 15 cells with "circular" morphology. These rounded cells appear brighter compared to other cells in the phase images. Typically, cells are assumed to adopt a circular morphology and become stationary before undergoing mitosis. The data set contains 12 cells with "protrusive" morphology. In the absence of live-imaging data for verification, these cells are assumed to be migratory with the presence of one or more lamellipodia on their cell boundary. The data set also contains 16 and 20 cells exhibiting "elliptical" and "elongated" morphology respectively. Elliptical cells are non-circular, oval or teardrop shaped and lack distinctive lamellipodia. Elongated cells are highly stretched along their principal axis. In other words, the maximum Feret length of elongated cells is much larger compared to their minimum Feret length.

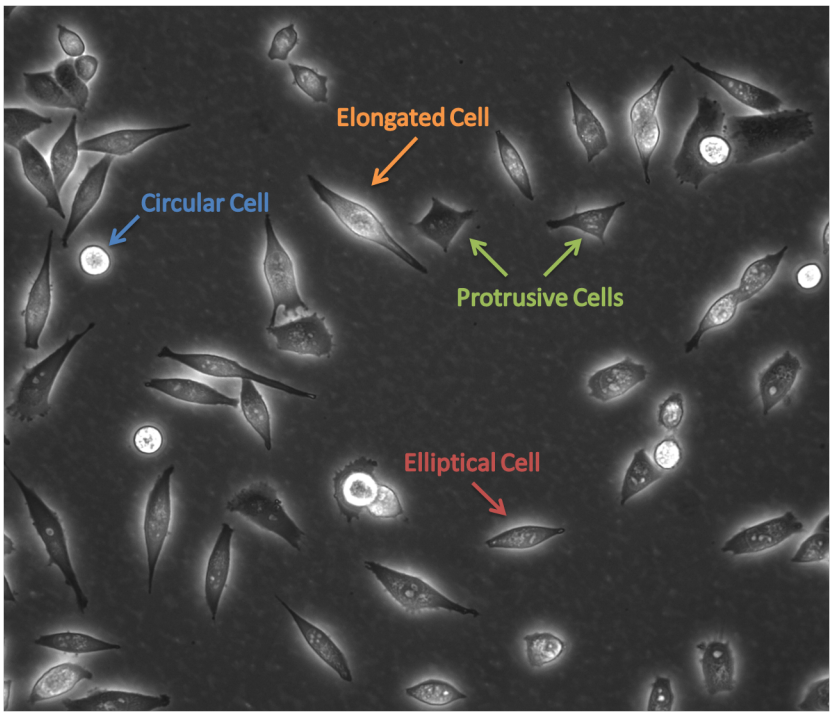


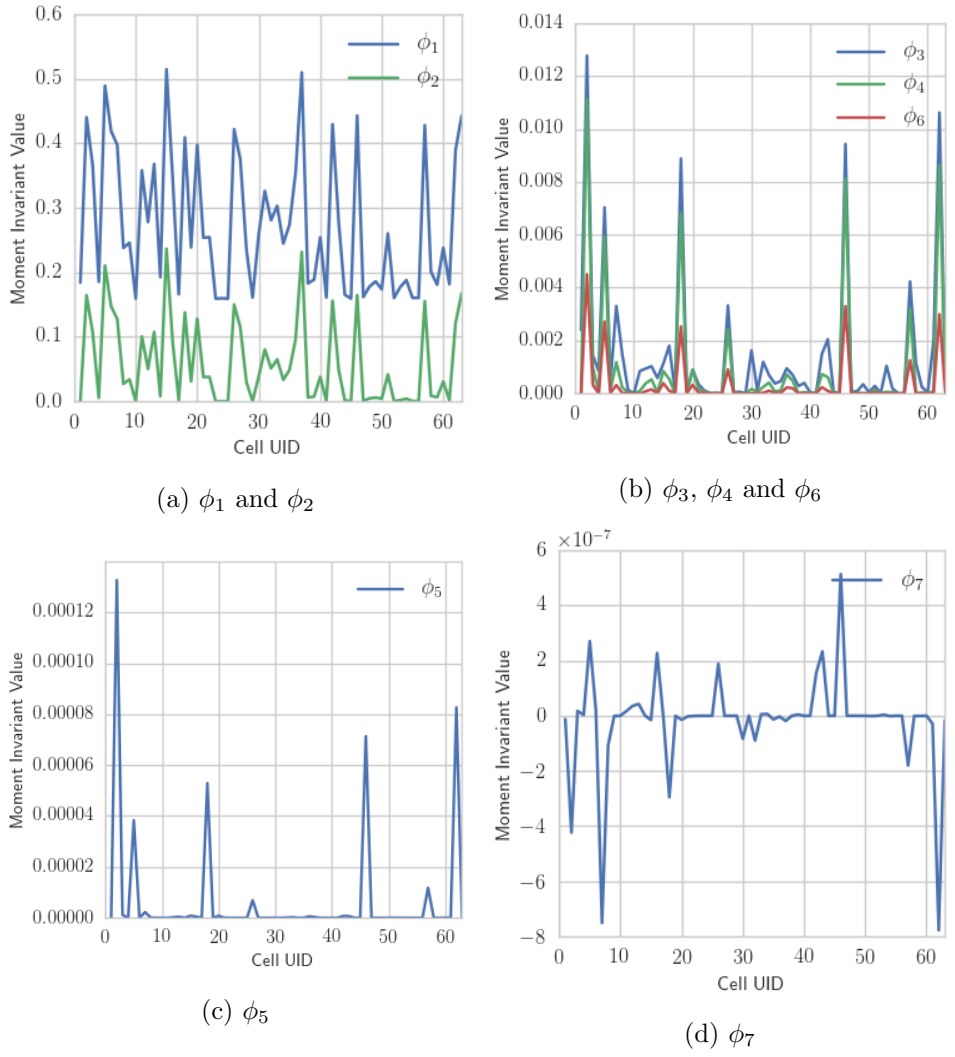
Figure 3.2: Four distinct morphologies in MIA PaCa-2 phase images

Validation of any unsupervised classification methodology requires comparison with ground truth. Ground truth refers to annotated data set where each data point is assigned a label based on the category to which it belongs. In this case, the data set consists of 63 cells, further categorized into four labeled groups. The four labels correspond to 12, 15, 16 and 20 cells with protrusive, circular, elliptical and elongated morphology respectively. The clustering algorithm also assigns labels (one label per identified cluster) to each data point. The efficacy of the methodology is determined by measuring the number of incorrect classifications. Labels assigned by the clustering algorithm are compared to ground truth labels in order to identify misclassified cells.

3.1 Exploratory Data Analysis

Unsupervised classification requires extraction of features from segmented cell images. A feature vector is computed for each cell using the feature extraction process described in Section 2.2. Feature extraction decomposes

cell images into their morphological characteristics using mathematical techniques to quantify cell shape and size. Each feature vector contains 27 features, including 7 Hu's invariant moments, 11 geometrical features, 3 boundary features and 6 shape factors. This section provides insight into diversity within the curated data set (consisting of 63 cells represented by points in the feature space) as well as relationships between different features.



(a) ϕ_1 and ϕ_2

(b) ϕ_3, ϕ_4 and ϕ_6

(c) ϕ_5

(d) ϕ_7

Figure 3.3: Plots of Hu's moment invariants for all cells in the data set

Hu's moment invariants are plotted against cell UID (arranged in no particular order) in Figure 3.3. Surprisingly, all moment invariants are correlated. The high degree of correlation is evident in Figure 3.3a, where ϕ_1 and ϕ_2 follow the same peak and trough pattern. Similarly, ϕ_3, ϕ_4, ϕ_5 (plotted separately in Figure 3.3c due to their low values) and ϕ_6 peak simultaneously for certain cells. Furthermore, ϕ_2, ϕ_3, ϕ_4, ϕ_6 and ϕ_5 increasingly lack resolution (in that order) and fail to provide any information that cannot be obtained from ϕ_1.

Rotationally symmetric objects have moment invariant values close to zero. Therefore, one can distinguish between circular cells and non-circular "stretched" cells (those that have a more elliptical or elongated morphology) by thresholding ϕ_1, as shown in Figures 3.4 and 3.5. Although cells with stretched morphology are typically larger in size in the MIA PaCa-2 data set, the discrimination is based on cell shape rather than cell size, as values of Hu's moment invariants are not affected by scaling cell size.

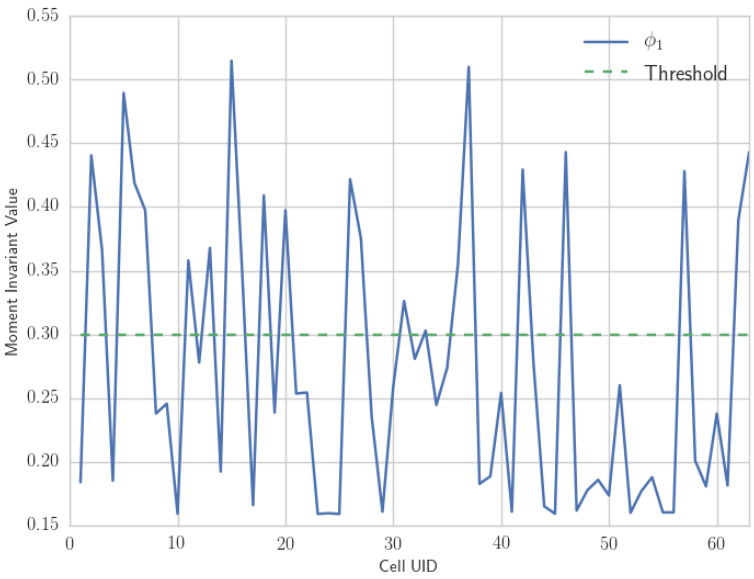
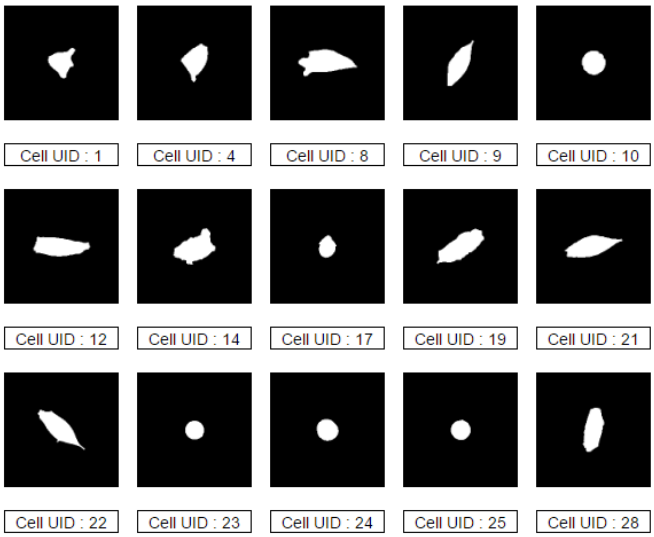


Figure 3.4: Thresholding Hu's invariant moment ϕ_1

The threshold value, 0.3, is determined by trial and error to separate cells into two groups as follows. Cells (identified by their UID on the horizontal axis) with circular morphology have ϕ_1 values below the threshold. Conversely, those with stretched morphology have ϕ_1 values above the threshold.

(a) Segmented cell images for subset of cells with $\phi_1 > 0.3$



(b) Segmented cell images for subset of cells with $\phi_1 < 0.3$

Figure 3.5: Manual classification of cells by thresholding ϕ_1

Skew invariants, ϕ_7 are plotted against cell UIDs in Figure 3.3d. Values of ϕ_7 occupy a narrow range from -8×10^{-7} to 6×10^{-7}. On first inspection,

it appears that one can classify cells into three distinct categories using two thresholds (see Figure 3.6). However, there is no clear visually apparent morphological difference between cells corresponding to positive peaks in ϕ_7 values and cells corresponding to negative peaks in ϕ_7 values, as shown in Figures 3.7a and 3.7b.

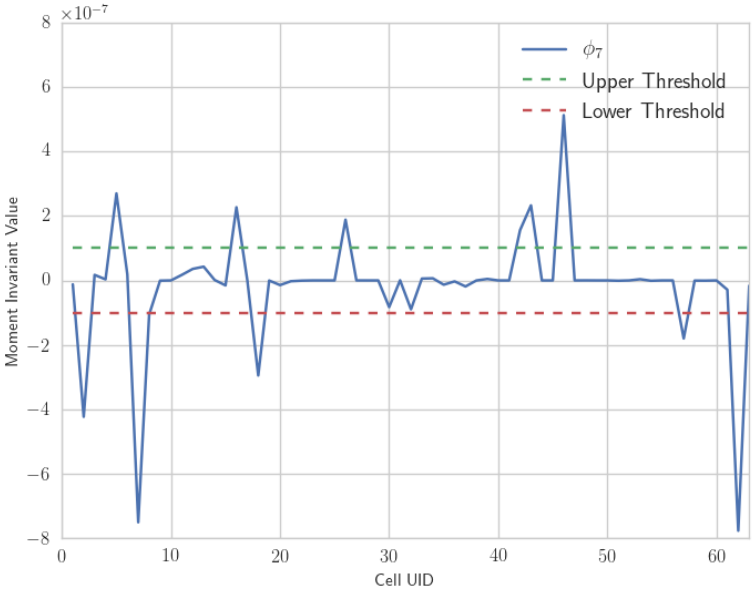
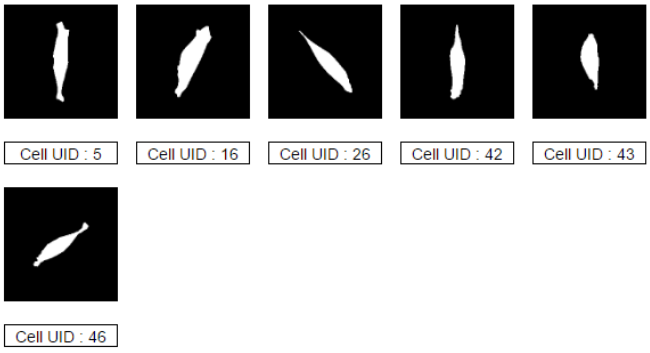
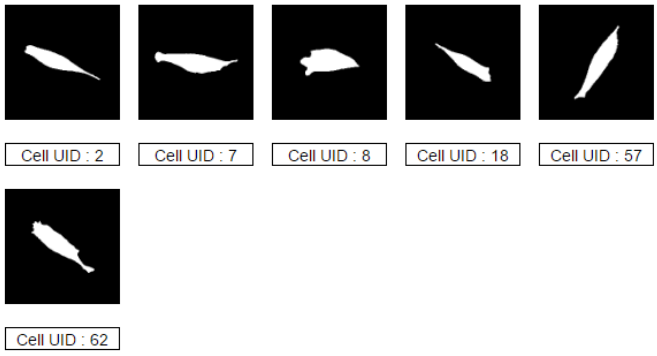


Figure 3.6: Thresholding Hu's invariant moment ϕ_7

Thresholds are arbitrarily chosen in an attempt to classify cells using ϕ_7. Cells with ϕ_7 values close to zero have circular morphology (not shown in this figure).



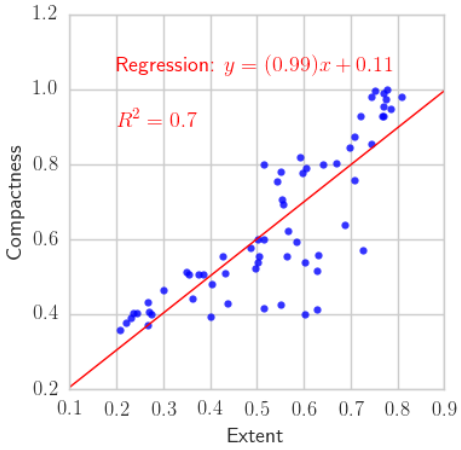
(a) Segmented cell images (and corresponding UIDs) for cells with ϕ_7 values above the upper threshold (corresponding to $\phi_7 > 10^{-7}$).



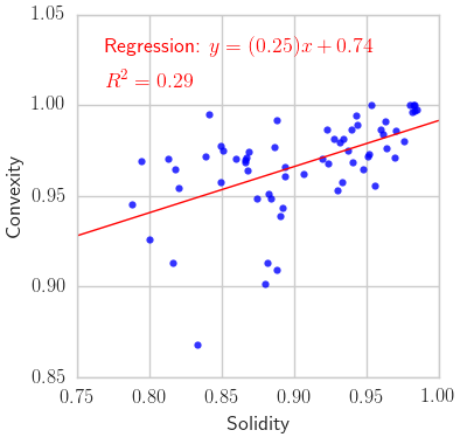
(b) Segmented cell images (and corresponding UIDs) for cells with ϕ_7 values below the lower threshold (corresponding to $\phi_7 < -10^{-7}$).

Figure 3.7: Manual classification of cells by thresholding ϕ_7

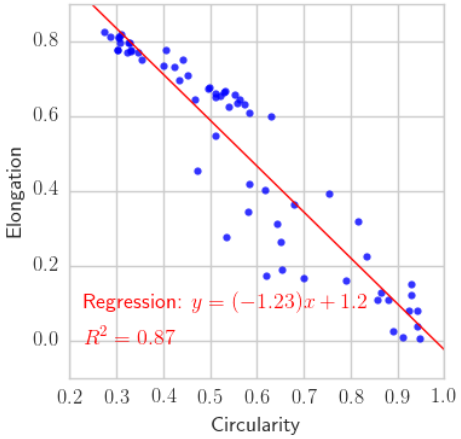
Pairs of shape factors, cell area and perimeter computed from segmented cell images are plotted against each other in Figure 3.8. Correlation between these features offers insight as to which features might be combined together during dimensionality reduction (PCA). Sum of squared residuals (R^2) quantifies the error in linear regression between pairs of shape factors.



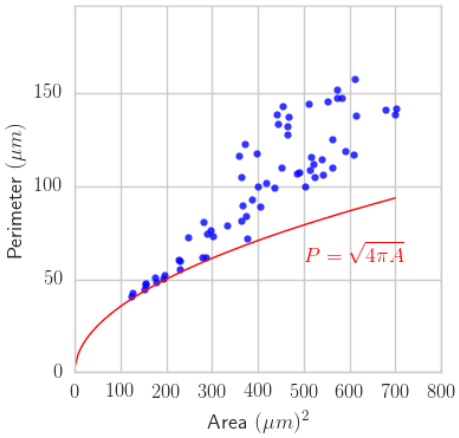
(a) Compactness vs. Extent



(b) Convexity vs. Solidity



(c) Elongation vs. Circularity



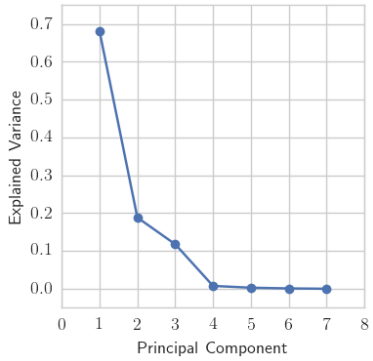
(d) Cell Area & Perimeter

Figure 3.8: Correlation between shape factors, cell area and perimeter

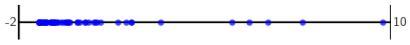
Strong positive correlation (evidenced by low R^2 value) is observed between convexity and solidity, as shown in Figure 3.8b. Note that the majority of cells in the data set are highly convex. The data set contains cells that represent values across a wide range of elongation and circularity. Circular cells can be identified by plotting perimeter versus area and identifying points that lie close to the curve $P = \sqrt{4\pi A}$ in Figure 3.8d. The more a cell deforms from circular shape, the further away from the curve will its (A, P) value be on this graph.

3.2 Clustering Using Hu's Moment Invariants

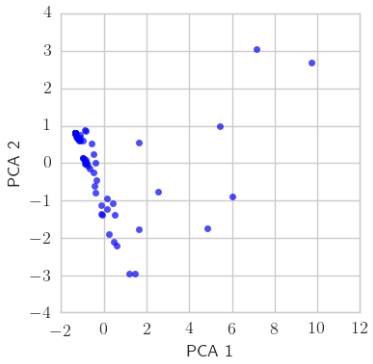
As demonstrated in Section 3.1, Hu's moment invariants can be used (with manually assigned thresholds) to distinguish between circular and stretched cell morphology. However, the goal of this thesis is to determine whether cells can be classified automatically. Can appropriate thresholds be determined implicitly and automatically by performing dimensionality reduction and clustering? To investigate further and answer similar questions regarding geometrical descriptors and shape factors in the following sections, the following procedure is implemented.



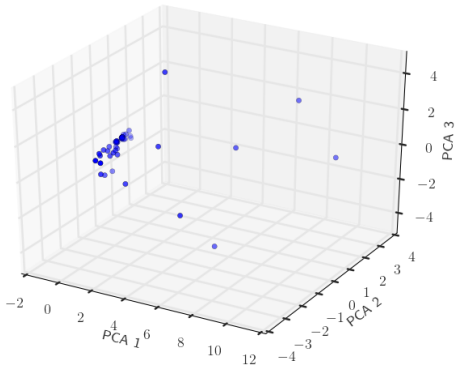
(a) PCA Elbow Plot



(b) 1-component PCA



(c) 2-component PCA



(d) 3-component PCA

Figure 3.9: PCA of Hu's moment invariants

Principal Component Analysis (PCA) is used to transform segmented cells denoted by points in seven dimensional feature space (consisting of all moment invariants, $\phi_1 \dots \phi_7$) to a lower dimensional subspace. Dimensionality reduction in this manner facilitates the identification of clusters using silhouette analysis.

According to the elbow plot (Figure 3.9a), two principal components account for over 80% of the explained variance. Since all moment invariants computed from MIA PaCa-2 segmented cell images happen to be correlated (see Section 3.1), most of the variance in the data can be captured by a single principal component, as shown in Figure 3.9b. This is further verified in 2-component and 3-component PCA plots (Figures 3.9c and 3.9d), where majority of the data points are clustered and outliers are responsible for the variance. However, in conformity with the rule of thumb (retaining over 80% of variance in data) and the elbow heuristic, two principal components are used in subsequent analysis.

The coefficients or weights assigned to the linear combination of features ($\phi_1 \dots \phi_7$) for the first and second principal component are (up to 2 decimal digits):

$$\mathbf{PC1} = (0.32, 0.322, 0.44, 0.44, 0.42, 0.44, -0.15),$$

$$\text{and, } \mathbf{PC2} = (-0.58, -0.58, 0.16, 0.15, 0.27, 0.13, -0.42),$$

respectively. By definition, majority of the variation in data is explained by the first principal component (**PC1**), represented by the horizontal axis in the 2-component PCA plot (Figure 3.9c). Projecting points on this axis provides an approximation of 1-component PCA, shown in Figure 3.9b.

Typically, this information is visualized in a biplot, like the one depicted in Figure 3.10. A biplot, in the context of PCA, consists of points and vectors drawn on a plot where the axes represent the principal components. Points are used to represent the transformed data set on the principal component axes, same as plots in Figure 3.9. Vectors, drawn on the same axes, represent the feature variables expressed in terms of the basis vectors of $(\text{PCA1}, 0)$ and $(0, \text{PCA2})$.

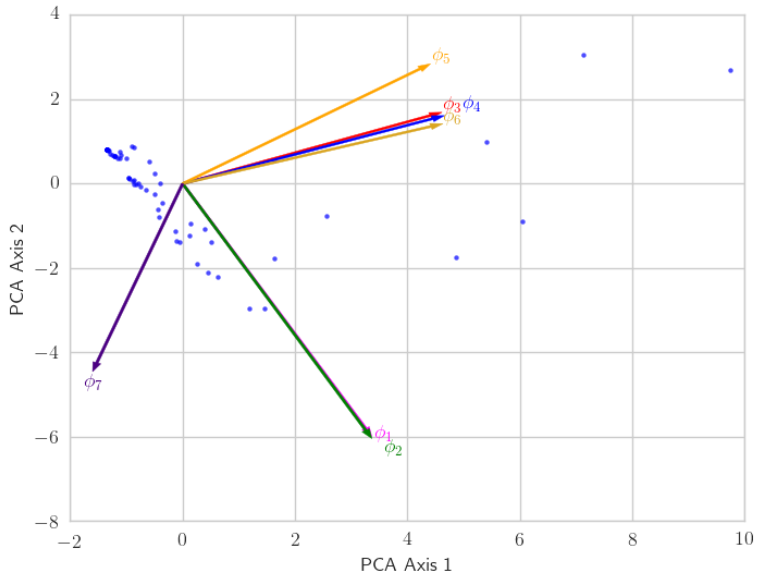


Figure 3.10: Biplot for 2-component PCA using Hu's moment invariants

The alignment of vectors in the biplot indicates their degree of correlation. Notice the correlation between ϕ_1 and ϕ_2, also evident in Figure 3.3a. Figure 3.10 suggests that if outliers (points scattered on the right) are removed, then majority of variation in the data will be captured in ϕ_1 and ϕ_2, as expected from the exploratory data analysis in Section 3.1.

Correlation between features can also be visualized with the aid of a feature agglomeration tree, shown in Figure 3.11. The tree is built bottom-up by recursively merging features (or combination thereof) using Ward's linkage method for hierarchical cluster analysis. Ward's method combines clusters of features using the sum of squared deviations from points to centroids as a distance metric. In other words, features are combined based on the proportion of variance explained by those features, in a manner similar to PCA. The leaves of the feature agglomeration tree are individual features in the feature vector. The number assigned to internal nodes indicates the degree of correlation between its children, with a higher number indicating greater correlation. The root of the tree is assigned the lowest number, since it combines two sub-trees (or clusters of features) with minimal correlation between them.

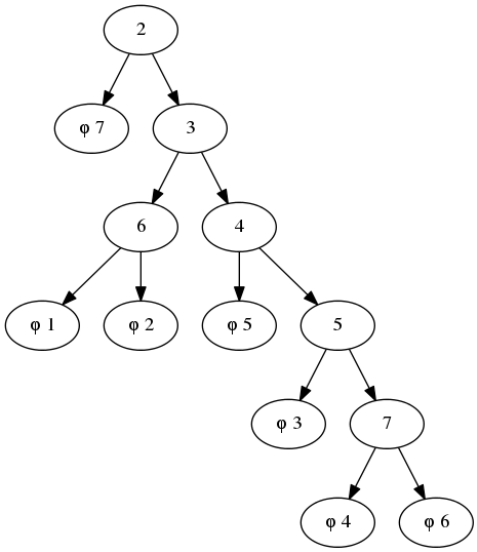
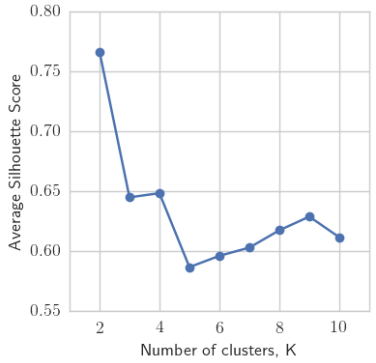


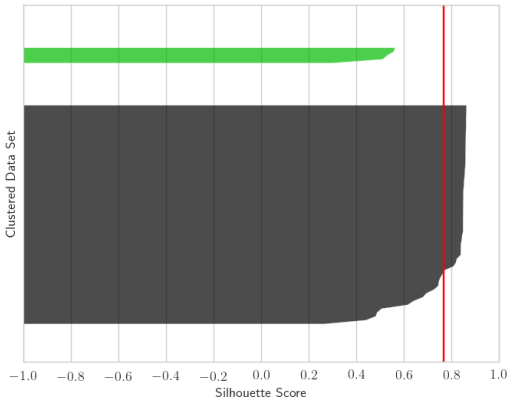
Figure 3.11: Feature agglomeration tree for Hu's moment invariants

Silhouette analysis using data points projected on two-component PCA axes predicted two clusters, as shown in Figure 3.12. Labeled data points are shown in Figure 3.12c. Same exact results, with regard to number of clusters and allocation of points to clusters, are obtained using 1-component or 3-component PCA.

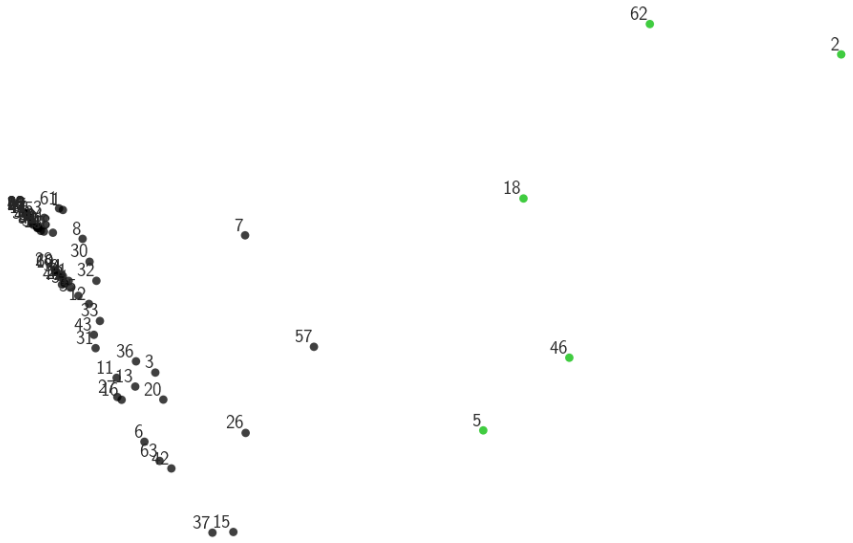
k-means clustering (with $k = 2$) using Hu's moment invariants as feature vector classified cells into two groups: 58 cells corresponding to black cluster labels and 5 cells corresponding to green cluster labels. Since the value of Hu's moment invariants is not affected by cell size, separation of cells in two clusters is based on cell shape. Cells labeled in green have morphology that is highly non-circular and corresponds to high values of Hu's moment invariants seen in Figures 3.4 and 3.5. Segmented cell images corresponding to labeled points in the clustering result are shown in Figure 3.13.



(a) Identifying number of clusters

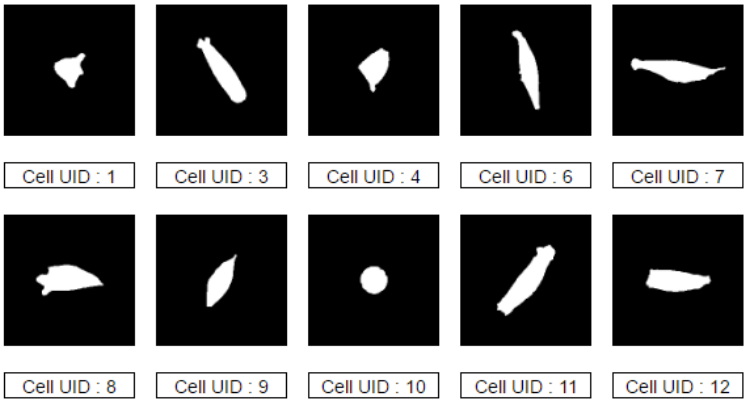


(b) Silhouette scores for $K = 2$

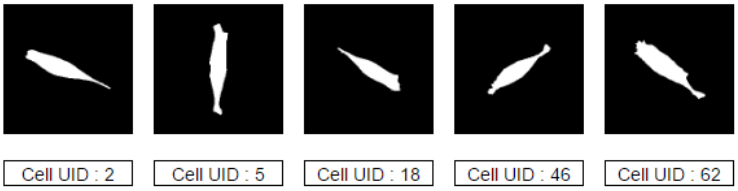


(c) Labeled data points (annotated with UID) for 2-component PCA

Figure 3.12: Silhouette score analysis of Hu's moment invariants



(a) Subset of cells corresponding to black cluster label

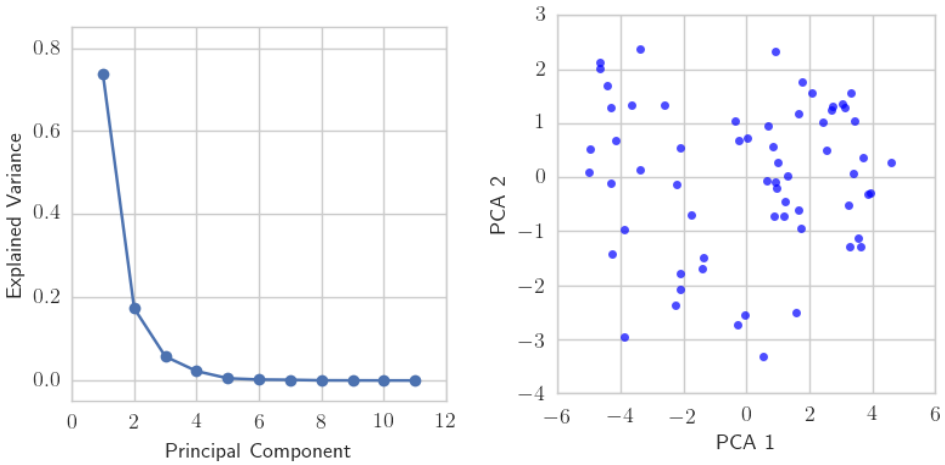


(b) All cells corresponding to green cluster label

Figure 3.13: Classification of cells using Hu's moment invariants

3.3 Clustering Using Geometrical Feature Descriptors

This section describes results obtained by following the same procedure as Section 3.2, but using geometrical features instead of Hu's moment invariants. Geometrical features (summarized in Tables 2.1 and 2.2) consist of information obtained from circle, ellipse, rectangle and polygon fits. PCA reveals that two principal components retain over 80% of explained variance, as shown in Figure 3.14a. Features expressed in terms of principal components are shown in the biplot (Figure 3.15). Ellipse perimeter and minimum Feret length are correlated and aligned with the second principal component. Majority of variance in the data (over 70% according to Figure 3.14a) is captured by the remaining features which are aligned in the direction of the first principal component. The feature agglomeration tree (Figure 3.16) confirms this observation.



(a) PCA Elbow Plot

(b) 2-component PCA

Figure 3.14: PCA of normalized geometrical features

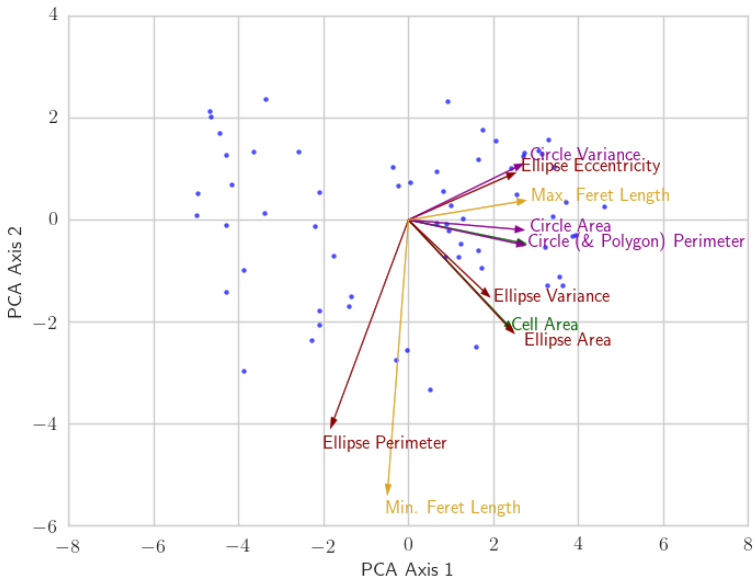


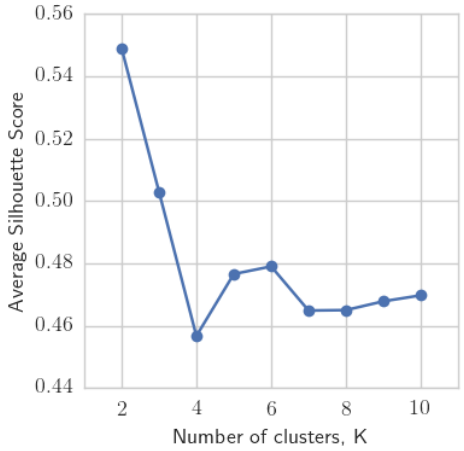
Figure 3.15: Biplot for 2-component PCA using geometrical features

Features obtained from ellipse fit, circle fit, rectangle fit and polygonal fit are plotted in red, magenta, yellow and green respectively.

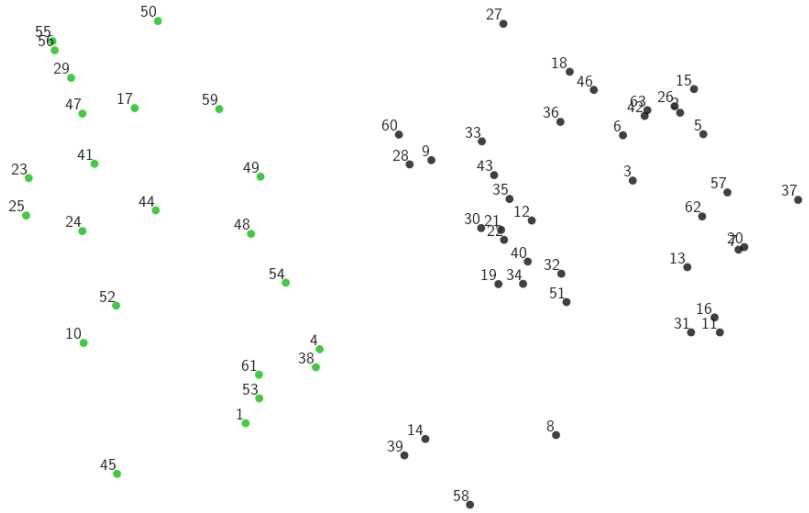
Figure 3.16: Feature agglomeration tree for geometrical features

Note that polygon area (denoted "Area" in graph above) and ellipse area are highly correlated. Similarly, polygon perimeter (denoted "Perimeter" in graph above) and circle perimeter are highly correlated. This is also noticeable in the alignment and (almost) equal magnitude of their vectors in the biplot (Figure 3.15), representing the (almost) equal weights assigned to these features in the principal components.

Two clusters are identified by performing silhouette analysis on geometrical feature data transformed using PCA, as shown in Figure 3.17a. The transformed data is clustered using k-means algorithm with parameter $k = 2$, to assign label to each cell UID (see Figure 3.17b).



(a) Identifying number of clusters using silhouette score



(b) Labeled data points (annotated with UID)

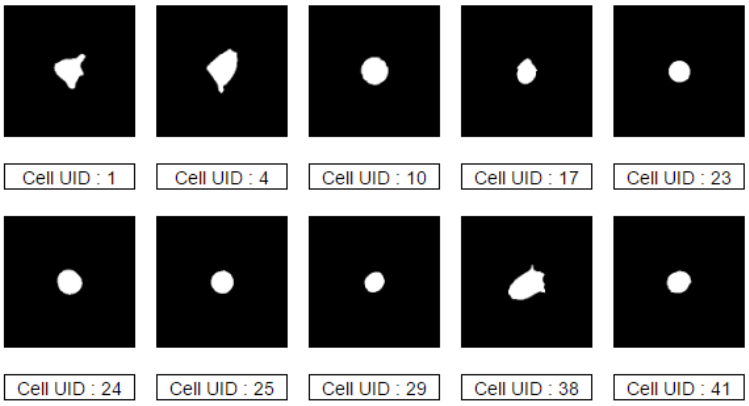
Figure 3.17: Silhouette analysis and clustering of geometrical features

Clustering using geometrical features resulted in improved classification of cells, clearly demonstrated by plotting segmented cell images corresponding to cluster labels in Figure 3.18. There is clear morphological difference between cells in Figure 3.18a and cells in Figure 3.18b. Cells corresponding to the 23 points in the green labeled cluster are smaller in size compared to cells corresponding to 40 points in the black labeled cluster. The classification is primarily based on cell size rather than cell shape, in contrast to classification using Hu's moment invariants. This can be easily verified, since some large cells with circular or protrusive morphology are labeled in black (see Figure 3.18c) as opposed to the majority of such cells that are labeled in green.

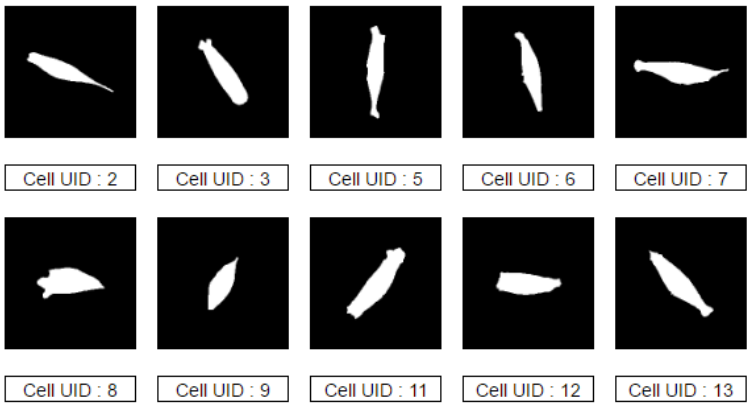
The results described above are obtained by classifying transformed features using k-means algorithm, using parameter $k = 2$ corresponding to the maximum silhouette score. However, if geometrical features are classified with $k = 4$ (not shown), using *a priori* knowledge that the data set contains four different morphologies, then morphologically similar cells are placed in the same cluster with only two mis-classifications. Consequently, employing a better cluster identification mechanism instead of silhouette score or replacing k-means with another clustering algorithm can improve unsupervised cell classification using geometrical features.

3.4 Clustering Using Geometrical and Boundary Features

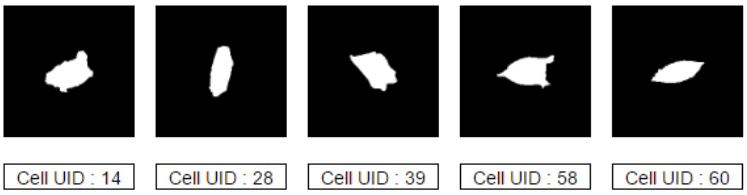
Another approach for improving results obtained in the previous section is to consider increasing variance in the data set by introducing additional features. Performing PCA on a larger set of features exploits increased variance and enables k-means to identify new clusters. Since geometrical features lack information about the boundary of cells, it is natural to assume that the combination of geometrical and boundary features will lead to better classification. Boundary features encode information about peaks and changes in the curvature of cell boundary. Therefore, boundary features can be used to distinguish between cells with multiple protrusions (i.e. more sign flips in curvature of boundary) and circular/elliptical cells that have mostly positive curvature. This hypothesis is tested in the this section.



(a) Subset of cells corresponding to green cluster label

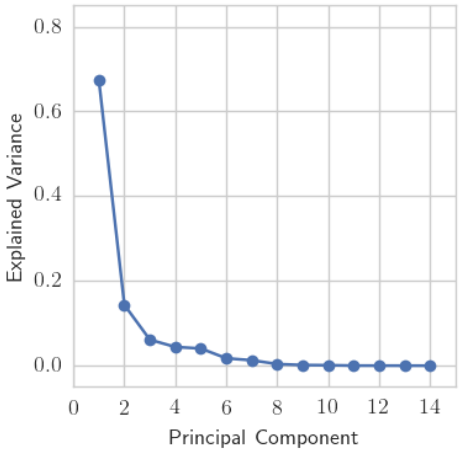


(b) Subset of cells corresponding to black cluster label

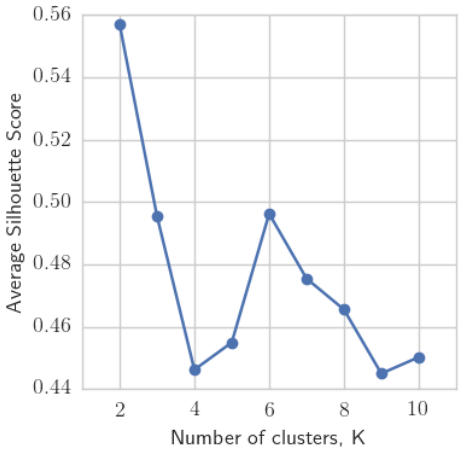


(c) Circular/protrusive cells in black cluster

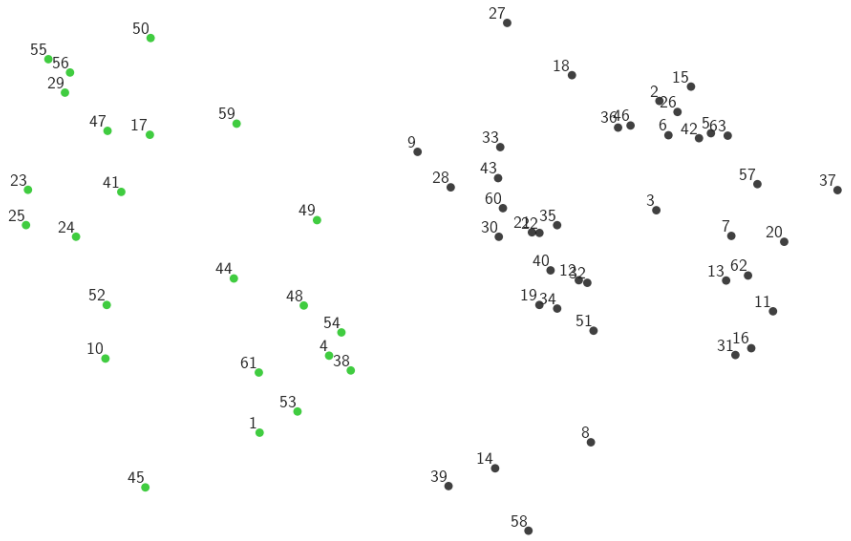
Figure 3.18: Classification of cells using geometrical features



(a) PCA Elbow Plot



(b) Identifying number of clusters



(c) Labeled data points (annotated with UID)

Figure 3.19: PCA, silhouette analysis and clustering of combined (geometrical and boundary) features

As shown in Figure 3.19, addition of boundary features did not lead to identification of any new clusters. In the two-component PCA space, placements of points in Figure 3.19c is almost identical to Figure 3.17b. The green cluster contains 40 points and the black cluster contains 30 points. There is no difference in the allocation of points to cluster labels. Possible reasons for lack of improvement can be ascertained by looking at correlations between features.

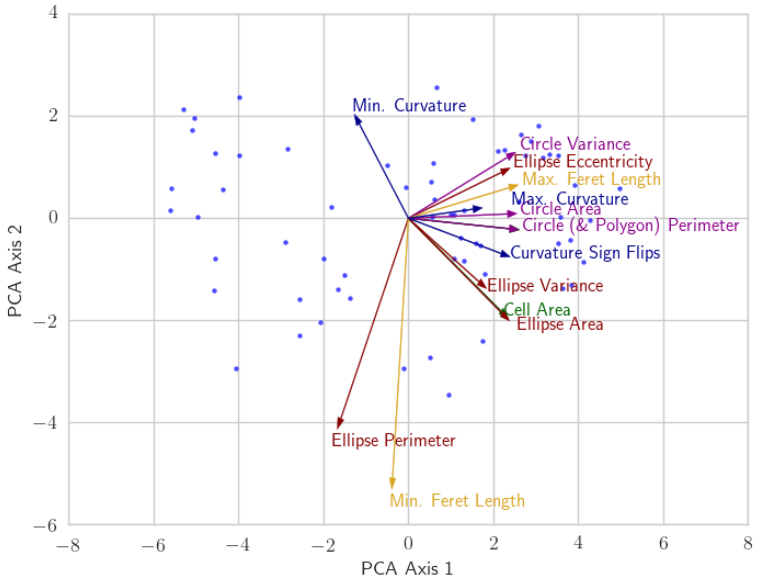


Figure 3.20: Biplot for 2-component PCA using combined features

The color scheme for geometrical features is same as Figure 3.15. Boundary features are plotted in blue.

Both biplot (Figure 3.20) and feature agglomeration tree (Figure 3.21) confirm that minimum boundary curvature is correlated to ellipse perimeter and minimum Feret perimeter. Maximum boundary curvature is correlated to ellipse variance and other features in the left sub-tree obtained by splitting the feature agglomeration tree at its root node.

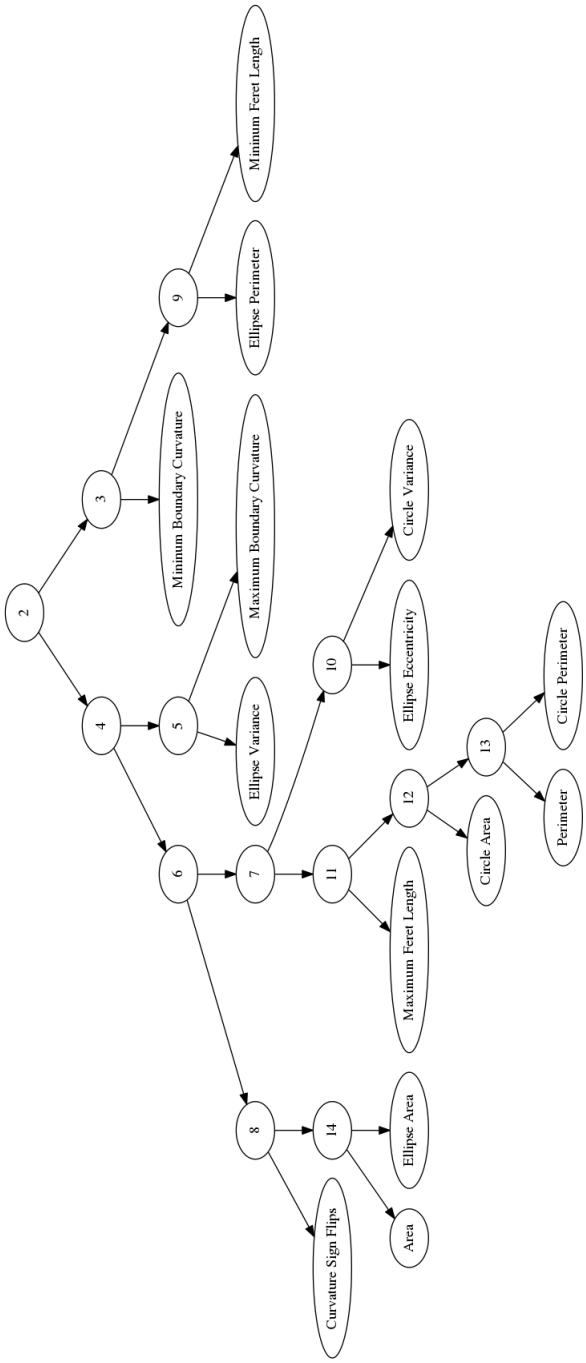
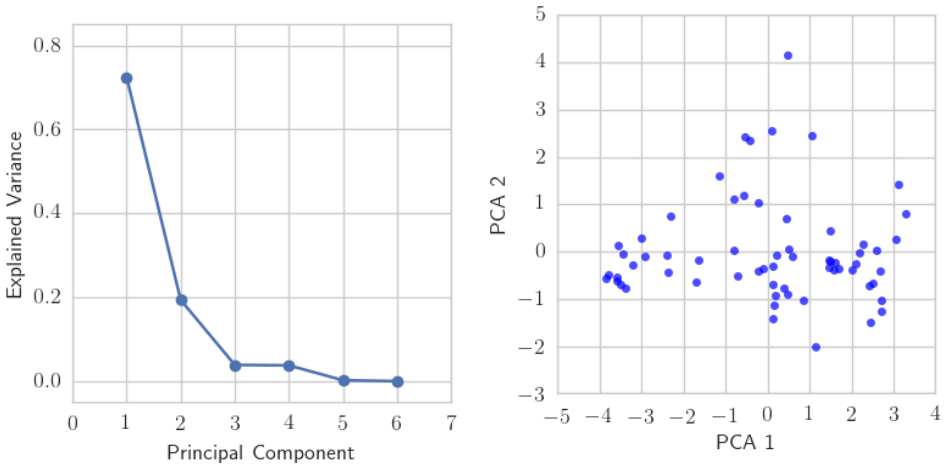


Figure 3.21: Feature agglomeration tree for combined features

3.5 Clustering Using Shape Factors

Six non-dimensional shape factors, extent, solidity, compactness, elongation, circularity and convexity (see Section 2.2.3 for definitions) were computed for each segmented cell image. Principal component analysis (Figure 3.22) reveals that over 80% of variance in data set can be captured using two principal components.



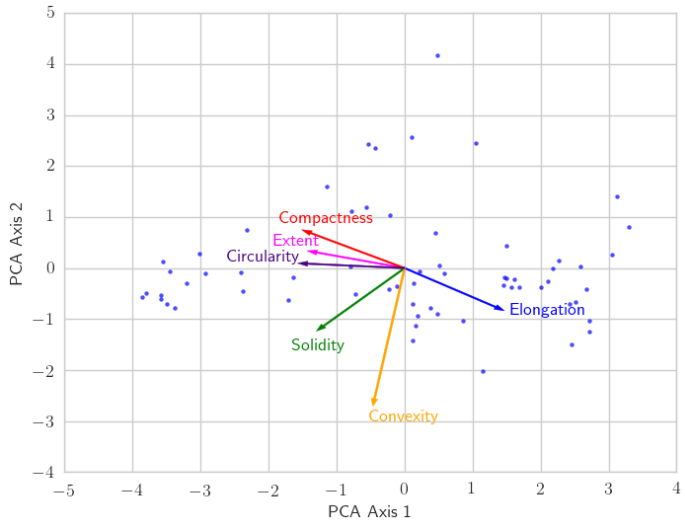
(a) PCA Elbow Plot

(b) 2-component PCA

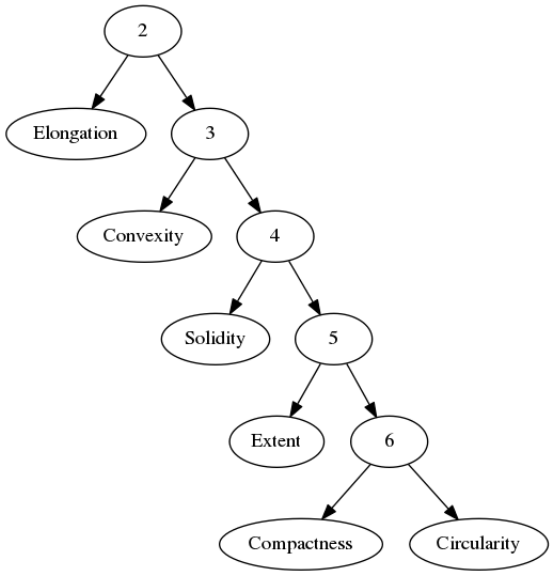
Figure 3.22: PCA of non-dimensional shape factors

Biplot diagram (Figure 3.23a) and feature agglomeration tree (Figure 3.23b) both confirm that compactness, extent and circularity are highly correlated. This indicates that similar clustering results (for the MIA PaCa-2 data set) can be obtained by computing just one out of these three shape factors.

Silhouette analysis confirms the presence of four clusters, as shown in Figure 3.24. Cell UIDs corresponding to the four cluster labels are shown in Figure 3.24c.

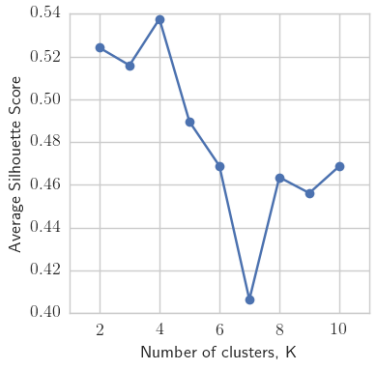


(a) Biplot for 2-component PCA using shape factors

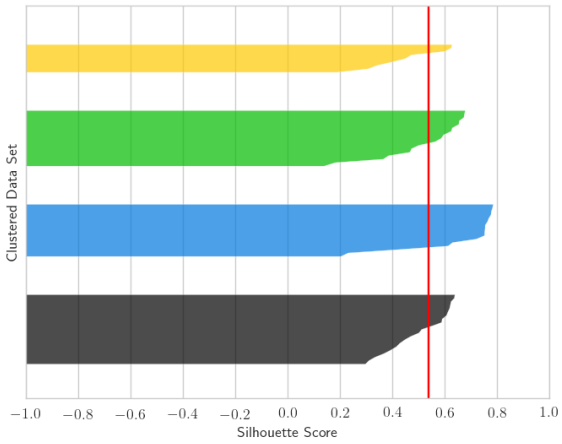


(b) Feature agglomeration tree for shape factors

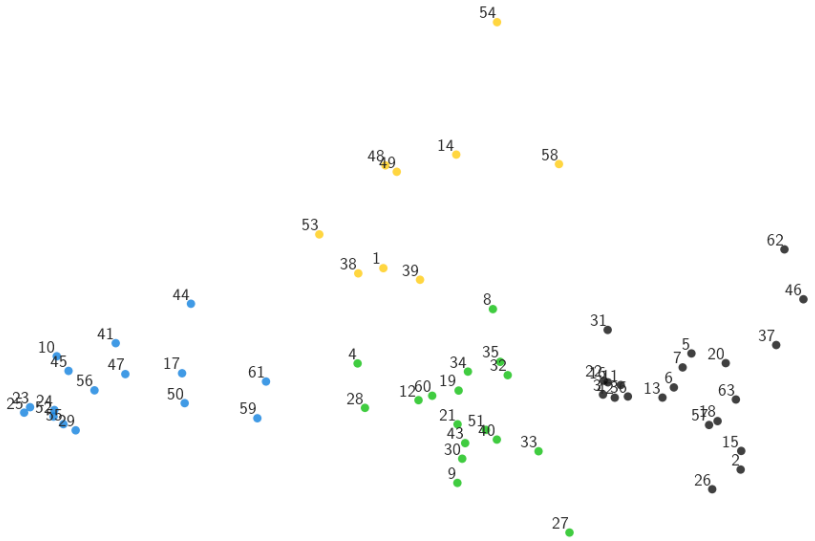
Figure 3.23: Analyzing correlation in shape factors using biplot and feature agglomeration



(a) Identifying number of clusters using silhouette score

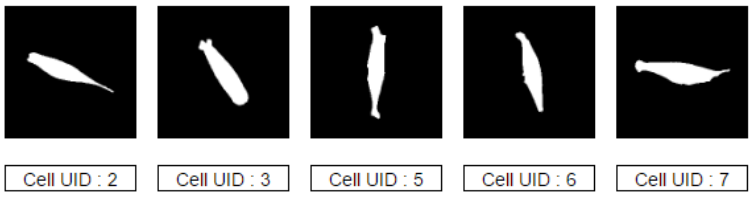


(b) Silhouette scores for each cluster

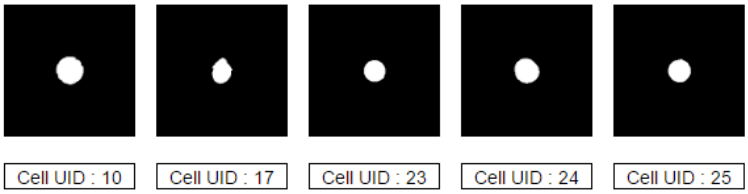


(c) Labeled data points (annotated with UID)

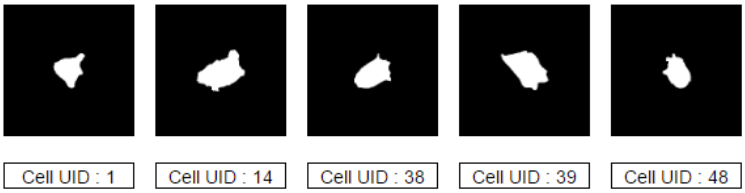
Figure 3.24: Silhouette analysis and clustering of shape factors



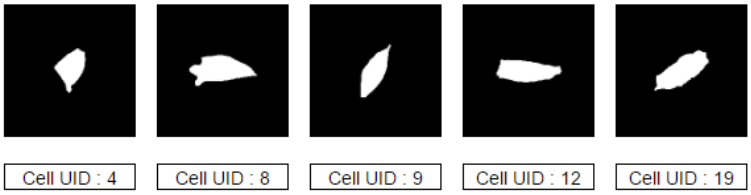
(a) Subset of cells corresponding to black cluster label



(b) Subset of cells corresponding to blue cluster label



(c) Subset of cells corresponding to yellow cluster label



(d) Subset of cells corresponding to green cluster label

Figure 3.25: Classification of cells using shape factors

Classification results are further improved using shape factors. A subset of cells corresponding to each cluster label are shown in Figure 3.25. The black, blue, yellow and green labels identify cells with predominantly elongated, circular, protrusive and elliptical morphology respectively. According to original annotations, four cells were mis-classified. An elliptical cell was mis-classified and incorrectly labeled as elongated. Three protrusive cells were

mis-classified: two were labeled as elliptical and one was labeled circular. Note that adding boundary features to the feature vector (in addition to shape factors) does not improve the result.

In summary, using shape factors as features led to automatic identification of four clusters corresponding to the elliptical, elongated, circular and protrusive morphologies evident in the manually vetted data set. Unsupervised classification of shape factors through a combination of PCA, silhouette score analysis and k-means resulted in just four mis-classifications. Hu's moment invariants, geometrical and boundary features did not perform as well as shape factors using this methodology. Classification of cells using Hu's moment invariants can be improved further by using t-SNE instead of PCA for dimensionality reduction. Results obtained using t-SNE are described in Appendix A.

Chapter 4

Conclusions

Correctly identifying specific objects such as cells in grayscale images with noise remains a challenging problem. Even the human visual system, that has evolved to recognize complex shapes within highly unstructured backgrounds can fail to find specific objects, misinterpret visual cues, or fail to detect cryptic shapes. What is more, even when all cells in a microscopy image are detected and outlined (segmented), it is still challenging to humans to distinguish between discrete classes when the classes have some overlap. For this reason, computational methods, which have nowhere near the discriminating power of the human visual system (at least for a small number of objects) are extremely challenging to develop. In this thesis, MIA PaCa-2 pancreatic carcinoma images are used to develop and test a pipeline for unsupervised cell classification, using a number of pre-existing methods that are adapted, improved, or modified. The process of assembling this pipeline has led to several areas of learning, both of the underlying biology, and of aspects of mathematical and computational aspects of the problems.

An image segmentation procedure that combines mathematical morphology and marker-based watershed segmentation algorithm is used to automatically obtain cell boundaries from phase-contrast images of MIA PaCa-2 cells. Correctly segmented cell boundaries are manually selected for feature extraction. The feature extraction process is not only an important component of the unsupervised classification pipeline, but it is also a means to quantify cell morphology in a manner that is useful for other applications. By constructing a feature vector through geometrical fitting, computation of boundary spline interpolation and shape factors as part of the overall methodology, the wealth of information gathered can be used by biologists to quantify cell and tissue morphology, determine the presence of multiple cell types, and/or mutations or epigenetic changes that affect cell shape. With additional development, the methods discussed here can be used to study changes in tissue geometry during morphogenesis using images acquired through time-lapse microscopy.

After feature extraction, a manually curated subset of cell features is used to identify circular, elliptical, protrusive and elongated cells. While the necessity for manual curation indicates some of the limitations of this methodology, it should be noted that MIA PaCa-2 cells do not exhibit a discrete set of shapes. As is the case with majority of other cell types, the morphology of MIA PaCa-2 cells lies in a continuum of shapes which makes their classification difficult, even for trained experts. For typical applications that require identification of distinct shapes in a heterogeneous population (arising from co-cultures, mixture of control and experimental group, etc.), variability in features is expected to be much higher compared to variability in the MIA PaCa-2 single cell line data. This suggests that one aspect of future work would be to test the methods on a variety of cell images, where there is a clearer distinction between cell types or morphologies.

Cell features are classified using the k-means clustering algorithm after performing dimensionality reduction to take advantage of correlation between features. PCA is used to compute a linear combination of features in order to project data from high dimensional feature space to low dimensional principal component space while retaining at least 80% of the variance in the original feature data. Silhouette analysis is used to determine the most probable number of clusters in the low dimensional space. After clustering, segmented cell images corresponding to cluster labels (assigned to cell UIDs) are used to identify the number of mis-classifications.

One of the goals for this thesis is to identify a minimal set of features that are computationally inexpensive to calculate but also competent in classifying cells. The number of mis-classifications is used to evaluate the performance of different kinds of features. As shown in Chapter 3, clustering using shape factors resulted in fewest mis-classifications. Improved cell classification using Hu's moment invariants is possible, although it requires the use of a non-linear embedding for dimensionality reduction (detailed in Appendix A). It should be noted that more information (besides global extrema and number of sign flips in curvature) can be obtained from the spline boundary fit, which may result in improvement of classification as well as identification of new classes based on locations of cell protrusions. Suggestions for improvement in the methodology and potential areas for future work are identified in the next chapter.

Chapter 5

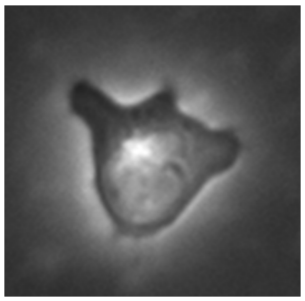
Future Work

The previous chapters describe a general methodology for unsupervised cell classification. The methodology is implemented in a modular pipeline that consists of algorithms used to perform image segmentation, feature extraction, dimensionality reduction and cluster identification techniques, in a sequential order. Each stage of the process takes input and produces output in a specified format, making the components of the pipeline replaceable. For example, the phase-contrast image segmentation component can be replaced by another component to segment images obtained through a different kind of microscopy. The pipeline is designed to handle high-throughput data, eliminating the need for manual intervention as much as possible. Currently, correctly segmented cells have to be manually identified and only a curated subset of the data is passed on to the dimensionality reduction step. With the improvements in methodology suggested below, it is possible that a fully automated pipeline will become a reality in the future.

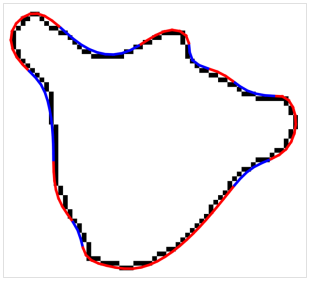
The addition of more quantifiable morphology based features can be used to increase the distance between non-similar cells represented by points in high dimensional feature space. This naturally improves the clustering of points after dimensionality reduction, potentially eliminating the need for manual curation. Suggestion for new features are highlighted below:

The curvature of the cubic spline interpolation of cell boundary identifies cell protrusions as regions of positive curvature, as demonstrated in Figure 5.1. However, for cells with long straight segments in cell boundary (see Figure 5.2), there are large number of changes in sign of curvature along the straight edges. Therefore, the number of positive curvature segments does not correspond to number of protrusions. As a result, the number of protrusions along the cell boundary is not represented in the boundary feature vector. Furthermore, information about the location of protrusions along the cell boundary is also missing from the boundary feature vector. This limits the methodology, preventing it from distinguishing between cells that

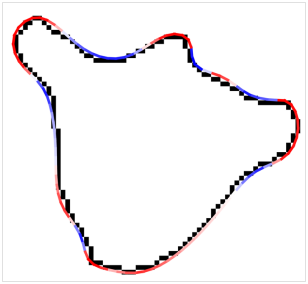
have protrusions oriented in the same direction and those cells that have random placed protrusions along their boundary.



(a) Cropped phase contrast microscopy image showing a protrusive MIA PaCa-2 cell

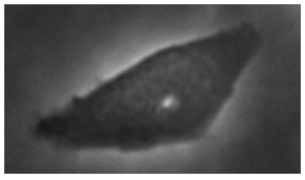


(b) Cubic spline interpolation of cell boundary showing regions of positive curvature (in red) and negative curvature (in blue)

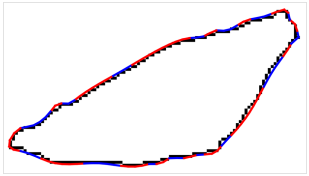


(c) Spline color-coded by magnitude of curvature, using the same color scheme as 5.1b

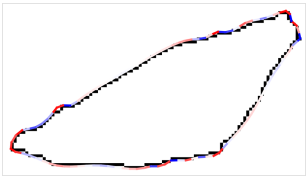
Figure 5.1: Boundary curvature of a protrusive cell



(a) Cropped phase contrast microscopy image showing an elliptical MIA PaCa-2 cell



(b) Cubic spline interpolation of cell boundary showing regions of positive curvature (in red) and negative curvature (in blue)



(c) Spline color-coded by magnitude of curvature, using the same color scheme as 5.1b

Figure 5.2: Boundary curvature of an elliptical cell

In addition to Hu's moment invariants, geometrical features, boundary features and shape factors, objects (including segmented cell images) can be classified by the degree of symmetry in their shape. In order to classify cells irrespective of their position and orientation, features should be invariant to image translation and rotation. Therefore, quantifying symmetry in a cell shape would require identifying its principal axes and computing features in relation to those axes. Furthermore, cell polarity (dissimilarity of "front" vs "back") is known to be an important aspect of cell motility. Future work

should aim to quantify this feature.

The choice of clustering algorithm and cluster identification strategy can significantly impact the outcome of unsupervised classification. While k-means and silhouette analysis are employed in this thesis due to their simplicity and widespread use elsewhere, replacing them with advanced clustering algorithms (e.g. OPTICS, variants of DBSCAN, Voronoi-based methods) will likely result in improved identification of clusters and robustness to noise. While two principal components captured over 80% of variance in data for all cases considered in Chapter 3, any choice of clustering algorithm should have the ability to deal with three or more principal components.

Methods for supervised classification of cells (based on morphological features) are not covered in this thesis for two reasons. Firstly, supervised classifiers like Support Vector Machines (SVMs), random forests and deep neural networks have already been shown to work for similar problems in recent publications (see Section 1.1). These methods require training using annotated data, a major disadvantage compared to unsupervised classification. Annotation of data is a laborious process that typically requires labeling cells experimentally (while making sure that the labeling technique does not have unintended consequences for structural and functional aspects of the cell) or manually assigning labels to each cell after image acquisition. Secondly, supervised classifiers like convolutional neural networks (CNNs) that do not require pre-computed features can easily achieve the end goal (i.e. cell classification based on morphology), but valuable quantitative information obtained as part of the feature extraction process is lost. CNNs belong to the class of pixel-based learning methods, which require a list of pixel intensities in the segmented cell image as input. Features are implicitly encoded in the weights and biases of neurons during the training process, thus eliminating the need for feature extraction. While a trained CNN can be used to identify distinguishing features in segmented cell images through deconvolution, the information thus obtained often lacks biophysical meaning.

Deconvolution of a CNN provides a list of image filters (a subset of which can be interpreted as edge and corner detectors) that can be applied to a segmented cell image in order to extract features for the purpose of classification. However, these filters typically do not correspond to measurable quantities like magnitude of curvature, perimeter length, etc. Despite the

lack of biophysical meaning, filters obtained by deconvolution might be used in conjunction with other feature extraction techniques described in Chapter 3 to yield improved classification results.

Certain artificial neural networks (ANNs) can be used in an unsupervised manner to perform feature extraction without training data. However, the features computed by these ANNs lack biophysical meaning, similar to CNNs. Autoencoders, a type of ANN, is widely used to perform dimensionality reduction. The efficacy of using an autoencoder network instead of PCA or t-SNE for the purpose of cell classification will be evaluated in the future.

The work described in this thesis is limited to the classification of cells in still images. By (re)-constructing the trajectory of cells and their lineage using images acquired through time-lapse microscopy, the feature vector can be augmented with information about cell movement and interactions between cells. Future work in this direction would involve computing "live imaging features" to study collective cell migration in wound-healing assays. The live imaging features will include cell velocity, neighbor count and estimate of cell cycle time. This would enable classification of cells based on migratory patterns and identification of cells that morph (from one shape to another) over time.

Bibliography

[1] Ilmari Ahonen, Ville Hrm, Hannu-Pekka Schukov, Matthias Nees, and Jaakko Nevalainen. Morphological Clustering of Cell Cultures Based on Size, Shape, and Texture Features. *Statistics in Biopharmaceutical Research*, 8(2):217–228, April 2016.

[2] Morteza Moradi Amin, Saeed Kermani, Ardeshir Talebi, and Mostafa Ghelich Oghli. Recognition of Acute Lymphoblastic Leukemia Cells in Microscopic Images Using K-Means Clustering and Support Vector Machine Classifier. *Journal of Medical Signals and Sensors*, 5(1):49–58, 2015.

[3] Richard Barnes, Clarence Lehman, and David Mulla. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers & Geosciences*, 62:117–127, 2014.

[4] Manuele Bicego and Pietro Lovato. A bioinformatics approach to 2d shape classification. *Computer Vision and Image Understanding*, 145:59–69, April 2016.

[5] R. G. Billiones, M. L. Tackx, and M. H. Daro. The geometric features, shape factors and fractal dimensions of suspended particulate matter in the Scheldt Estuary (Belgium). *Estuarine, Coastal and Shelf Science*, 48(3):293–305, 1999.

[6] Christina Carlsson. *Vehicle size and orientation estimation using geometric fitting*. PhD thesis, Division of Automatic Control, Department of Electrical Engineering, Linkpings universitet, Linkping, 2000. OCLC: 474207954.

[7] Thanatip Chankong, Nipon Theera-Umpon, and Sansanee Auephanwiriyakul. Automatic cervical cell segmentation and classification in Pap smears. *Computer Methods and Programs in Biomedicine*, 113(2):539–556, February 2014.

[8] D. Chaudhuri and A. Samal. A simple method for fitting of bounding rectangle to closed regions. *Pattern Recognition*, 40(7):1981–1989, July 2007.

[9] 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(1), August 2016.

[10] S. Dimopoulos, C. E. Mayer, F. Rudolf, and J. Stelling. Accurate cell segmentation in microscopy images using membrane patterns. *Bioinformatics*, 30(18):2644–2651, September 2014.

[11] G. A. Dunn and A. F. Brown. Alignment of fibroblasts on grooved surfaces described by a simple geometric transformation. *Journal of cell science*, 83(1):313–340, 1986.

[12] P. Foggia, G. Percannella, C. Sansone, and M. Vento. Benchmarking graph-based clustering algorithms. *Image and Vision Computing*, 27(7):979 – 988, 2009. 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007).

[13] Hubert Fonga. Pattern recognition in gray-level images by Fourier analysis. *Pattern Recognition Letters*, 17(14):1477–1489, December 1996.

[14] Rui Gradiz, Henriqueta C. Silva, Lina Carvalho, Maria Filomena Botelho, and Anabela Mota-Pinto. MIA PaCa-2 and PANC-1 pancreas ductal adenocarcinoma cell lines with neuroendocrine differentiation and somatostatin receptors. *Scientific Reports*, 6(1), April 2016.

[15] Radim Halr and Jan Flusser. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, volume 98, pages 125–132. Citeseer, 1998.

[16] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[17] Zhihu Huang and Jinsong Leng. Analysis of Hu's moment invariants on image scaling and rotation. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 7, pages V7–476. IEEE, 2010.

[18] K. Inoue and K. Kimura. A method for calculating the perimeter of objects for automatic recognition of circular defects. *NDT International*, 20(4):225–230, August 1987.

[19] Ferenc Kovcs, Csaba Legny, and Attila Babos. Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.

[20] David J. Logan, Jing Shan, Sangeeta N. Bhatia, and Anne E. Carpenter. Quantifying co-cultured cell phenotypes in high-throughput using pixel-based classification. *Methods*, 96:6–11, March 2016.

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[22] Erik Meijering. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, 2012.

[23] L. A. Merson-Davies and F. C. Odds. A morphology index for characterization of cell shape in Candida albicans. *Microbiology*, 135(11):3143–3152, 1989.

[24] Fernand Meyer. The watershed concept and its use in segmentation: a brief history. *arXiv preprint arXiv:1202.0216*, 2012.

[25] Loris Nanni, Michelangelo Paci, Florentino Luciano Caetano dos Santos, Heli Skottman, Kati Juuti-Uusitalo, and Jari Hyttinen. Texture Descriptors Ensembles Enable Image-Based Classification of Maturation of Human Stem Cell-Derived Retinal Pigmented Epithelium. *PLOS ONE*, 11(2):e0149399, February 2016.

[26] Ryusuke Nosaka and Kazuhiro Fukui. HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns. *Pattern Recognition*, 47(7):2428–2436, July 2014.

[27] Eric Olson. Particle shape factors and their use in image analysis-part 1: Theory. *Journal of GXP Compliance*, 15(3):85, 2011.

[28] Fred Park. Shape descriptor/feature extraction techniques, 2011.

[29] Yaling Pei and Osmar Zaane. A synthetic data generator for clustering and outlier analysis. 2006.

[30] Richard J Prokop and Anthony P Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, September 1992.

[31] Lorenzo Putzu and Cecilia Di Ruberto. White blood cells identification and classification from leukemic blood image. In *Proceedings of the IWBBIO international work-conference on bioinformatics and biomedical engineering*, pages 99–106, 2013.

[32] Y. Qin, W. Wang, W. Liu, and N. Yuan. Extended-Maxima Transform Watershed Segmentation Algorithm for Touching Corn Kernels. *Advances in Mechanical Engineering*, 5(0):268046–268046, January 2015.

[33] Carolina Reta, Leopoldo Altamirano, Jesus A. Gonzalez, Raquel Diaz-Hernandez, Hayde Peregrina, Ivan Olmos, Jose E. Alonso, and Ruben Lobato. Segmentation and Classification of Bone Marrow Cells Images Using Contextual Information for Medical Diagnosis of Acute Leukemias. *PLoS ONE*, 10(6), June 2015.

[34] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[35] Christoph Sommer and Daniel W. Gerlich. Machine learning in cell biology teaching computers to recognize phenotypes. *Journal of Cell Science*, 126(24):5529–5539, December 2013.

[36] H. Q. Sun and Y. J. Luo. Adaptive watershed segmentation of binary particle image. *Journal of Microscopy*, 233(2):326–330, February 2009.

[37] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson Addison Wesley, 2006. Google-Books-ID: YHsWngEACAAJ.

[38] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, January 2001.

[39] Shutong Tse, Laura Bradbury, Justin WL Wan, Haig Djambazian, Robert Sladek, and Thomas Hudson. A combined watershed and level set method for segmentation of brightfield cell images. In *SPIE Medical*

Imaging, pages 72593G–72593G. International Society for Optics and Photonics, 2009.

[40] Cristina Urdiales, Antonio Bandera, and F. Sandoval. Non-parametric planar shape representation based on adaptive curvature functions. *Pattern Recognition*, 35(1):43–53, 2002.

[41] Xavier Vasques, Laurent Vanel, Guillaume Villette, and Laura Cif. Morphological Neuron Classification Using Machine Learning. *Frontiers in Neuroanatomy*, 10, November 2016.

[42] A. A. Yunis, G. K. Arimura, and D. J. Russin. Human pancreatic carcinoma (MIA PaCa-2) in continuous culture: sensitivity to asparaginase. *International Journal of Cancer*, 19(1):128–135, January 1977.

Appendix A

Dimensionality Reduction Using t-SNE

t-distributed stochastic neighborhood embedding (t-SNE) is a dimensionality reduction technique that produces a non-linear embedding from high dimensional space to low dimensional space. t-SNE is often used in place of PCA due to its tendency to preserve local structure in data. Unlike PCA, t-SNE is a non-parametric learning algorithm that handles non-linearity in the data very well. The embedding is learned in the process of moving data to the low dimensional space. Consequently, t-SNE does not provide a function for transforming data from the high dimensional space to the low dimensional space. Furthermore, the t-SNE algorithm requires multiple input parameters including perplexity, early exaggeration, learning rate and number of iterations. While default values of these parameters work well for widely publicized open data sets, the algorithm is sensitive to perplexity and learning rate parameters for features included in the MIA PaCa-2 data set. The perplexity parameter is similar to k in the k-nearest neighbors (KNN) classifier algorithm. It is used to build a nearest neighbor graph in the high dimensional feature space. The t-SNE model building process involves performing random walks on this feature graph. The learning rate parameter plays an important role in preventing the algorithm from getting stuck in a local minimum while minimizing the Kullback-Leibler divergence, a non-convex cost function. For more details on the t-SNE algorithm, please refer to Maaten and Hinton (2008) [21].

Shortcomings of t-SNE, including its stochastic nature (requiring multiple runs to ensure convergence), absence of parameter estimation techniques and lack of simplicity (compared to PCA where the linear transformation can be easily analyzed), are the main reasons for its exclusion from the main text of this thesis.

As shown in Chapter 3, Figure 3.12c, Hu's moment invariants did not cluster properly using PCA. However, using 2-component t-SNE, four clusters corresponding to circular, protrusive, elliptical and elongated cells are easily identifiable (see Figures A.1 and A.2). The results described in this appendix are limited to 2-component t-SNE to allow for comparisons with results obtained by using PCA.

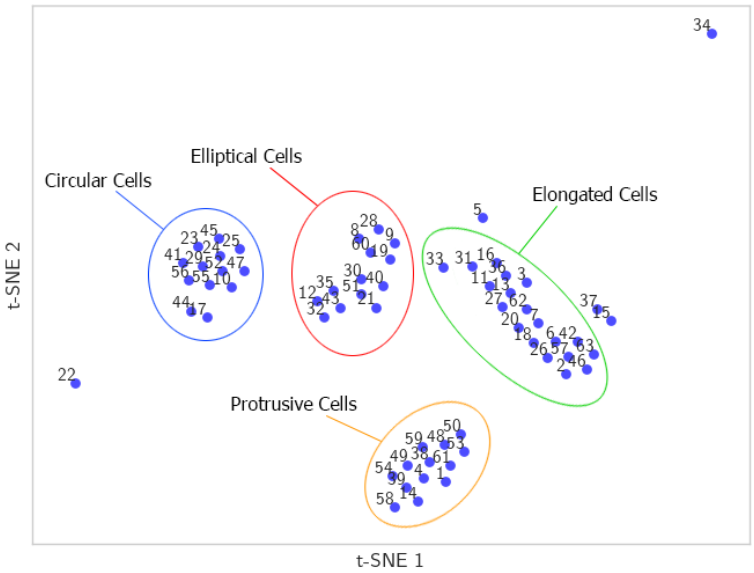
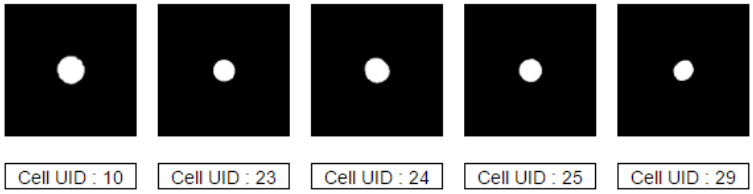
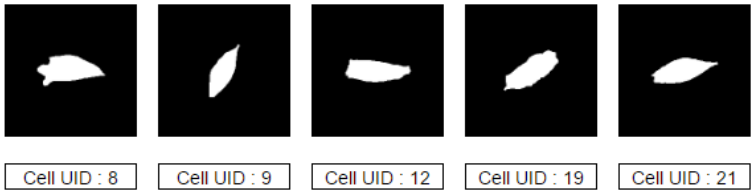


Figure A.1: Clustering Hu's moment invariants using 2-component t-SNE

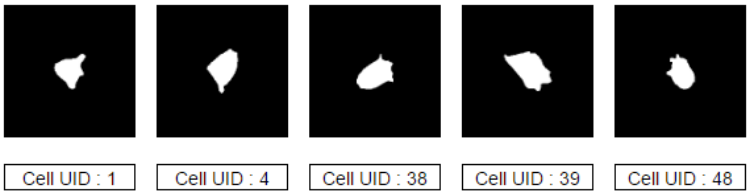
13 circular cells, 13 elliptical cells, 13 protrusive cells, 19 elongated cells and 5 outliers are identified in the 2-component t-SNE plot. Contrary to observations in Section 3.1, a non-linear transformation of Hu's moment invariants is capable of distinguishing between various cell morphologies. However, finding parameters for the t-SNE algorithm (perplexity = 10 and learning rate = 500) requires manual exploration of the parameter space. Furthermore, if new data is made available, then a new embedding has to be learned in order to transform the data into the two dimensional space.



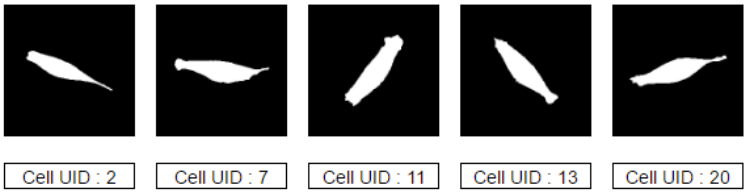
(a) Subset of circular cells clustered within blue region



(b) Subset of elliptical cells clustered within red region



(c) Subset of protrusive cells clustered within yellow region



(d) Subset of elongated cells clustered within green region

Figure A.2: Improved classification of cells using Hu's moment invariants

Clustering is not evident when combination of geometrical features and boundary features are embedded in two-dimensional space using t-SNE, as shown in A.3. A similar t-SNE plot of (only) geometrical features (not included here) has nearly identical placement of points. Thus, adding boundary information to geometrical features does not result in improvement of unsupervised classification, reinforcing the result obtained using PCA.

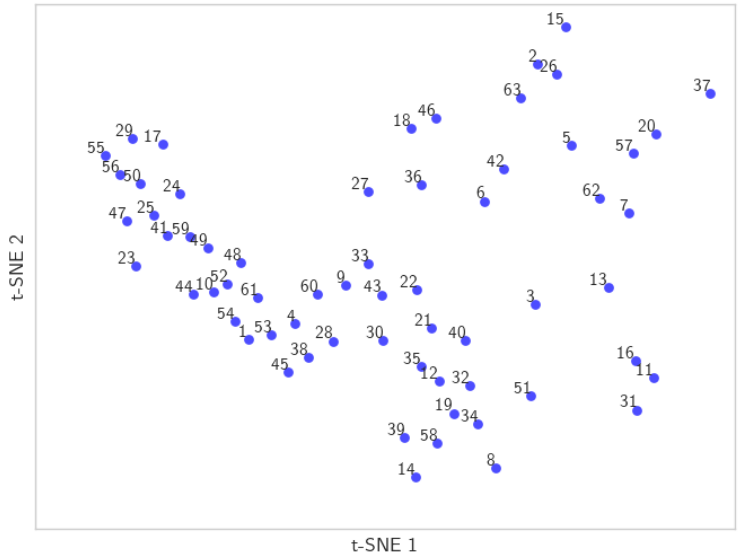


Figure A.3: t-SNE using combination of geometrical and boundary features

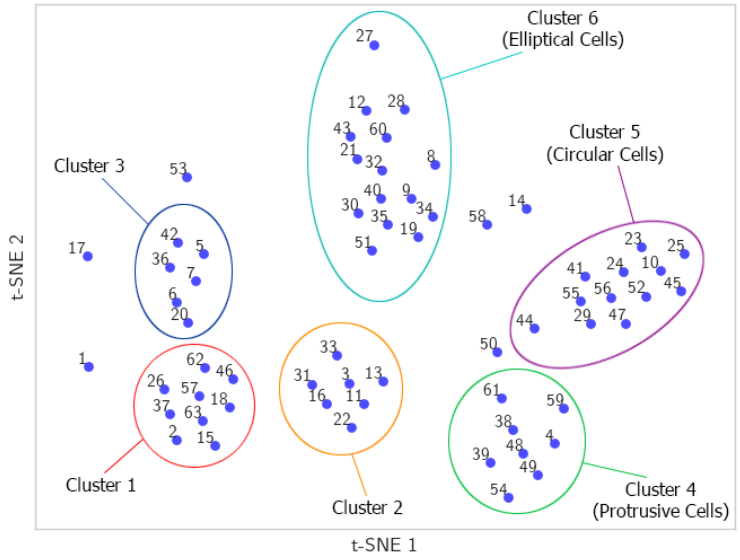
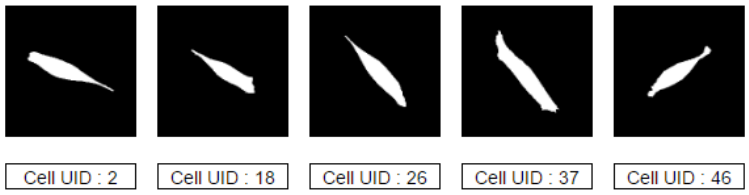


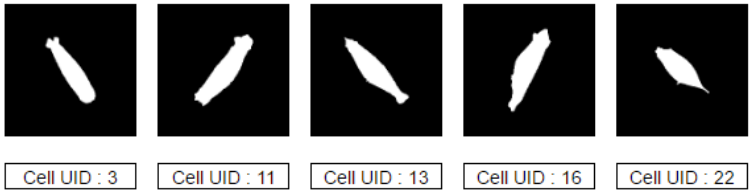
Figure A.4: Clustering shape factors using 2-component t-SNE

Six clustered regions are identified in the two component t-SNE plot of shape factors, as shown in Figure A.4. Clusters 1, 2 and 3 correspond to

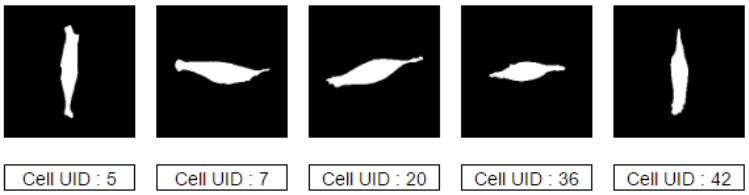
a more elongated morphology (see Figure A.5), while clusters 4, 5 and 6 correspond to protrusive, circular and elliptical cells respectively (see Figure A.6). One can argue that some of the outlier points (see Figure A.7) correspond to cells with somewhat ambiguous morphology.



(a) Cluster 1



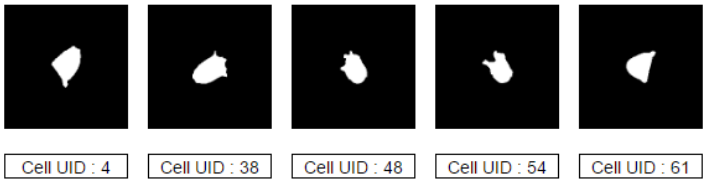
(b) Cluster 2



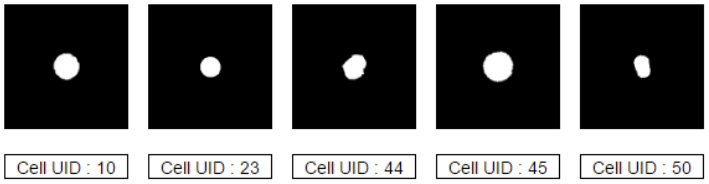
(c) Cluster 3

Figure A.5: Subset of elongated cells corresponding to clusters 1, 2 and 3

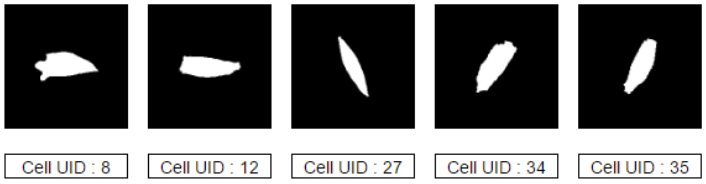
Notice the subtle distinction in morphology between different clusters of elongated cells in Figure A.5.



(a) Subset of protrusive cells corresponding to Cluster 4



(b) Subset of circular cells corresponding to Cluster 5



(c) Subset of elliptical cells corresponding to Cluster 6

Figure A.6: Protrusive, circular and elongated cells identified in Figure A.4

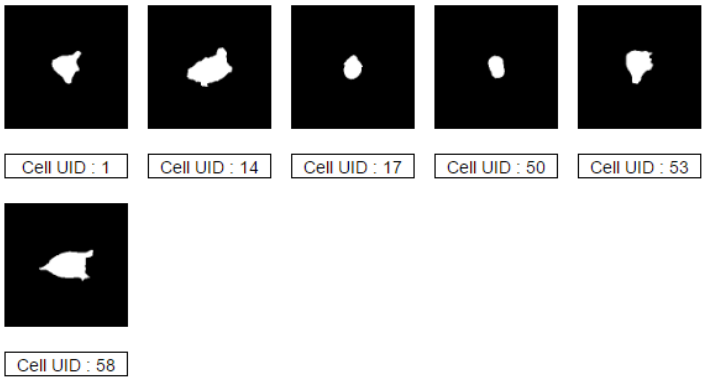


Figure A.7: Segmented cell images of outlier points in Figure A.4