CLASSIFICATION AND DETECTION OF CELL BASED ON MORPHOLOGY

by

Raj Patel

A project report submitted in conformity with the requirements
for the degree of Master of Science, Information Technology

Department of Mathematical and Physical Sciences
Faculty of Graduate Studies
Concordia University of Edmonton

CLASSIFICATION AND DETECTION OF CELL BASED ON MORPHOLOGICAL SHAPES

RAJ PATEL

Approved:

Supervisor : Baidya Nath Saha, Ph. D. Date

Committee Member Date

Dean of Graduate Studies: Alison Yacyshyn, Ph. D. Date

CLASSIFICATION AND DETECTION OF CELL BASED ON MORPHOLOGY

Raj Patel
Master of Science, Information Technology

Department of Mathematical and Physical Sciences
Concordia University of Edmonton
2022

Abstract

Every living species has cells and based on those cells scientists observe and make some predictions or observations. The identification and classification of cells is a highly crucial part of medical research and involves human efforts to a huge extent. Due to human involvement, the research may go wrong and may predict inaccurate results. This time-consuming process can be eliminated if machine learning algorithms is applied. This research aims to train an ML algorithm to detect and classify cells based on a number of branches. This research will take images of a group of cells as input to the algorithm for training purposes. The training of the model will be accomplished based on the labeled images of cells. The trained model will identify and classify the cells images in three different classes Class A, Class B, and Class C. Class A cells will not have any branches, Class B will have fewer branches and Class C will include cells with many branches. To eliminate the inaccurate results.The training will be done by using Machine Learning algorithm which is used for classification problems. The training part will involve extracting features of cell from the image of the dataset. As a result of this research, medical practitioners will get the classified images of cells and they will be able to analyze the result in an organized manner. The trained model will be helpful to provide better and more accurate results. This will fasten the process of analyzing medical reports. This model can be enhanced to classify the cells and also analyze the classification report to provide better predictions regarding the cells.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The study of cell shape, structure, form, and size is known as cell morphology. Cell morphology in bacteriology refers to the size and form of bacteria: cocci, bacilli, spiral, and so on, and most human cells produced in culture may be classified into three types depending on their morphology: Fibroblasts, epithelial-like cells, and lymphoblasts are all types of cells. Cellular morphogenesis is critical in developmental biology. The cell's surroundings has a major impact on it, and its reaction to biophysical and topographical stimuli is mediated by mechanosensing, mechanotransduction, and mechanoresponse. Cells extracted from multicellular structures (tissues, organs) and cultivated as monolayers change their morphology from spherical to spindle-like, elongated forms, according to research. Morphological traits play an important part in cancer detection, with normal cells having regular, ellipsoid forms and malignant cells being irregular and curved. Cell morphology has also been linked to cell mobility and, eventually, tumour invasiveness. The dynamic morphology of migratory cells is extremely complicated, necessitating the use of cutting-edge tools to analyse and characterise it.[1]

Recent cell biology research has revealed a proclivity to investigate cellular morphology at the single-cell level, since individual morphology must be considered to properly identify various cell types governed by cellular material, active features, and external environment. As a result, morphology is commonly utilised as a metric of cell categorization. Various microscopy methods might be used to acquire cellular morphology, offering information at various levels. Fluorescence microscopy, in particular, has transformed the area of cellular imaging research. Fluorescence imaging approaches, on the other hand, need the use of fluorescent dyes and fluorescent probes, which are not accessible for all target molecules.[2]

1.2 Problem Statement

Microscopy is a popular technique for examining cells grown in conventional culture conditions. Traditional methods of analysis involve the manual, subjective classification of cells based on their morphology or form, which can be linked to certain functional functions, viability statuses, or other valuable outputs. Manual categorization, on the other hand, is both extremely subjective (it is dependent on the skill level, experience, and viewpoint of the trained observer) and time consuming. With technological advancements, very high throughput gathering of microscopic pictures is now available, which regrettably might overwhelm an individual's ability to categorise cells within a dataset.

The goal of this research is to create an automated classifier using Machine Learning that can be used to identify cells based on their observable shape and enable for the quick and impartial measurement of various cellular morphologies. This method will allow for the quick examination of high-throughput microscopy data without the subjectivity of a human classifier, hence enhancing the throughput and reliability of morphological categorization.

Although the main challenge is that there is not data except the co-ordinates of x and y axis with classification mainly A,B,C only for 3 images with its images and its very challenging to process the images which has moderate to high noise and light interference in images with cell size being one case. Still after developing a automated algorithm for detecting and analyzing the cell this will increase the progress time of a project to increase many fold because manual classification takes months whereas automated takes maximum one day and it will give all outputs including number of cell in a image and also how many are there in each categories or class.

1.3 Contribution of the thesis

- The project is based on images gathered and provided by an expert. The expert has given 888 'tif' extension images and 12 images with manual classification, binary images and outlines images(drawn outline on cell using ImageJ). Though only one image is useful that is the original image. Which has different cell of different morphology.

- First of all, the images that are gathered does not have specific data that can be used to train and test the machine learning algorithm. All the cells in a particular images is detected or predicted by the algorithm and converted to binary images.

- Using machine learning algorithm, feature extraction was performed on each cell of a distinct images and every detail of a single cell from that image was extracted and training and testing data was created.

- After the cells have been subdivided, the next step is to represent each of them with a collection of characteristics. The features are summed together into a feature vector that is sent into the supervised machine learning classifier. After being trained on labelled data, the classifier can discriminate between a predefined set of cell classifications

- Machine Learning classifiers were run on a subset of the data, known as the train data, to generate a model that can be used to evaluate the remaining data. The models were created and the accuracies of each model were assessed to see which model performed the best. KNeighbors Classifier,Support Vector Machine, Gaussian Process Classifier, Classifier,Decision Tree, Random Forest Classifier, Ada Boost Classifier, Linear Discriminant, MLP Classifier, Quadratic Discriminant,Logistic Regression, Gradient Boosting Classifier, LGBM Classifier, an are used.

- For supervised learning, accuracy and precision were measured for each classifier. The goal feature is known for supervised learning, which aids in training the remaining data. The data with the target variable is used to train the model. The model is tested on the test data, which generates the target automatically.

1.4 Organization of the thesis

This research project contain six(6) chapters. Chapter 1 is the Introduction part which explains the background of the research, the problem for which this research is made, that is, the detection and classification of cell based on morphology using Machine Learning algorithm. Chapter 2 is Literature Review in which the cell detection of classification of different research based on different approaches is mention in this section and how this research are related tho this thesis.

Chapter 3 is Methodology section which is divided into three section which are Cell Detection, Feature Extraction and Cell Classification. Each section describe the steps taken to develop this project in a sequential manner.

Chapter 4 is Data Description,it explains how data is gathered, what is the source of data and the quality of data.What kind of images are used for data training and testing.

Chapter 5 show the experimental results and discussion of this whole thesis, what kind of problem where faced when applying different approaches during execution of this project, what are the results and how much accurate the model made is.

Chapter 6 concludes the thesis works and what kind of other features and approaches can be use in the future to make this project more successful and provide accurate output.

Chapter 2

Literature Review

Individual cells change their shape in response to environmental stimuli and selection forces, as well as their differentiation state. While the vast majority of these cues and pressures are known to be mediated by changes in intracellular signal transduction, the particular regulatory mechanisms that regulate cell shape, size, and polarity remain unknown. Systematic cell morphology research includes manipulating metabolic pathways and detecting phenotypic changes. To make this process easier, experimental biologists require software that can analyse a huge number of microscopic pictures in order to classify cells and detect cell kinds [3].Mostly, the challenge is to work with nuclear image because its too small and sometimes its take the cell as noise.Not only, detection is difficult but classification is not easy, as the inaccuracy of the algorithm is high because every cell needs to be classify differently.There are many researcher who develop code on cell classification and detection according to their particular dataset which might fit for other cell but classification and accuracy needs to developed based on nature of cell.

Input data for cell classification is often a randomised list of feature vectors, where each feature vector is an ordered set of "Features or Descriptors" that determine a cell's morphology. The remainder of this section discusses current morphology-based research.

2.1 Machine Learning based approaches

A major percentage of contemporary cell categorization material is devoted to leukaemia diagnosis. Putzu *et al* [4]. segmented lymphocyte nuclei from blood samples to identify whether an individuals have acute lymphoblastic leukaemia.They employed a Support Vector Machine to conduct binary classification and calculated 30 shape descriptors, 4 colour descriptors, and 16 texture descriptors to characterise abnormalities in nuclear shape. Patients with this kind of disease were classified with 0.25 percent accuracy.Also, the techniques use has similarity with this research.

Based on the French-American-British (FAB) [3] classification system, *Amin et al* improved the ALL classification approach by calculating a broader collection of geometrical and statistical parameters from blood and bone marrow smears to further categorise cell nuclei into three subtypes: L1, L2, and L3. Soon after, *Reta et al* used a number of classifiers to differentiate between acute leukaemia families (ALL and AML), ALL subtypes (L1 and L2), and AML subtypes (M2, M3, and M5) [5].This classification is very similar to cell classification in this project.

Nanni *et al* used ensembles of cell texture descriptors to categorise phase-contrast microscopy pictures of retinal pigment epithelial (RPE) cells (produced from human pluripotent stem cells) [6]. They use 3 image descriptors for this, first is based on preprocessing of images, second is region based and third is based on teture and featue of images this methods gives them results on human cell in details.

Dhananjay Bhaskar[3] provides a method for identifying cells in microscope pictures and calculating quantitative descriptors to define their morphology.Though his use of feature of cell for classification is same as this thesis but they overlap images to reduce dimensionality without decreasing accuracy.

Stringer *et al* [7] employ Cellpose, a deep learning-based segmentation approach that can precisely separate cells from a wide range of picture formats while requiring no model retraining or parameter tweaks.It can detect cell in Three dimension(3-D) and two dimension(2-D), where use of 3D is done by stacking multiple layer of 2D which eliminate the use of 3D labeled data.

2.2 Phase Contrast based Approaches

In terms of cellular morphology of reconstructed pictures, Phase contrast generates artefacts between closely placed cells with clearly discernible borders. Reconstruction with Phase Contrast algorithm results in an even backdrop with no visible artefacts around the cells, although cell boundaries are absent and mitotic cells are not accurately recreated [8].This type of image reconstruction is very useful for this thesis images where cell needed to constructed with its edges for branches for discerning.

2.3 Threshold Based Cell Segmentation

The Simple threshold (sST) outperforms automatic thresholding strategies such as the Poisson distribution (sPT) or the Otsu method (sOtsu). The power of these automated algorithms rests in picture segmentation, where the ideal threshold value differs between images. This is not required for QPI photos or images constructed by removing background error. The main advantage of the sOtsu and sPT approaches is that there are no parameters to optimise. Morphological changes could potentially enhance thresholding outcomes. In terms of processing times, these are the simplest and thus fastest approaches, which are provided mostly to provide a fundamental understanding of our segmented data [8].Although the automatic thresholding is fast but only with images that have no background or by removing background.But this method failed as this data has very high noise ratio.

2.4 Graph Cut based Segmentation

Based on Graph-Cut, there are numerous ways and adaptations. We simply tested the basic model in this article. When Graph-cut was applied to the reconstructed images, it earned the highest Dice coefficient among non-trainable techniques, save for ChanVese. Graph-Cut, on the other hand, does not significantly outperform basic threshold, offering only a 2% boost in Dice coefficient and is only acceptable for rebuilt data. In terms of distinctions amongst microscopic approaches, the Graph-cut strategy, followed by PC and HMC, was most suited for reconstructed DIC pictures. In terms of computational times, this strategy outperforms level-set-based solutions [8].

2.5 Single Cell Perceptron Segmentation

After reconstruction, foreground segmentation, and seed-point extraction, the data was segmented using Marker-controlled watershed (MCWS) using distance transforms or images directly. Errors caused by this stage have only a little impact on overall segmentation quality when compared to prior steps, delivering few-pixel shifts to one or more adjacent cells. The distance transform approach is more general, but if the cells are well separated, the MCWS-only approach may yield better results [8].There is one approach on single cell prerceptron where there is use of single layer neural network for detection and finding of cell with borders based on pixel.

Chapter 3

Cell Detection and Classification

This thesis, explains a methodology for supervised classification of cells from a phase contrast microscopy images. In this, the images will be put into different machine learning algorithms for detection and segmentation, extraction of cell features, classification of cell in 3 types(that is A,B and C) and validation steps, which are explained in details below.

When the output of the machine learning model is a real or continuous variable, regressions are performed.

3.1 Cell Detection

For long years, image analysis has struggled with nucleus segmentation in 3D and 2D. Labeling and segmenting cells and subcellular structures like nuclei has been a focus of research for decades. Many biological disciplines have a requirement to find, count, and segment nuclei and nuclear markers for quantification in a variety of investigations [9]. Except for neuron segmentation, the great majority of existing segmentation methods are based on a few fundamental approaches: intensity thresholding, feature extraction, morphological filtering, area accumulation, and deformable model fitting [10].

By mis-specificating the cell region to be split or over-segmenting, region accumulation methods such as Voronoi-based approaches or watershed transform might result in erroneous cell borders. Similarly, popular deformable model approaches, such

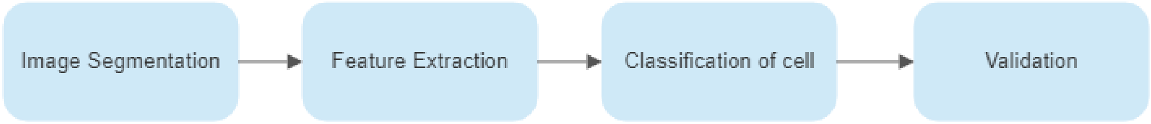
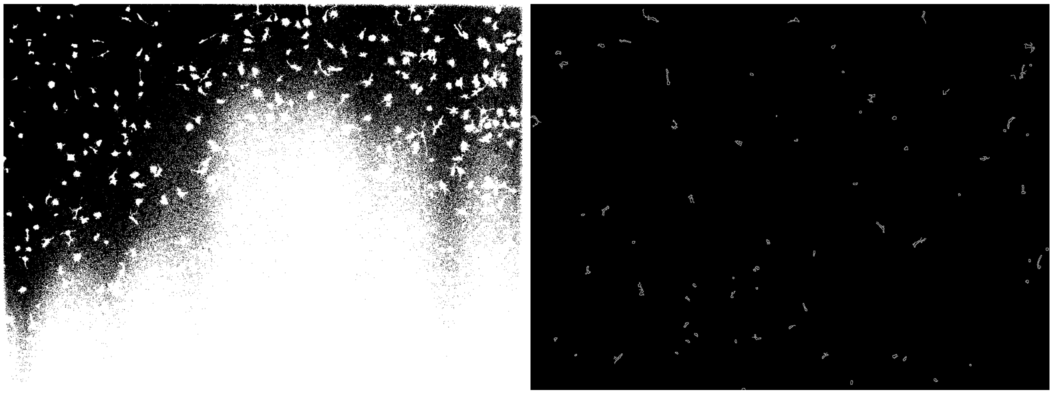


Figure 3.1: Standard steps to classify cell of microscopy image

as geodesic active contours or level sets, which detect cell boundaries by minimising functional energy, the result which is in poor boundary detection because of its uses local optimization algorithms that only guarantee to find a local minimum or decode the boundary information using the gradient vector field of the image [3], [10]. Although use of otsu and canny edge detection with other methods failed in this data set because of high noise and light interference as shown in figure 3.2.



(a) Otsu Thresholding

(b) Canny edge detection

Figure 3.2: Failure while detecting cell due to high noise and light interference

In figure 3.2,during microscopy there was light or shado in the images which lead to results in image A, also image B (canny edge detection) cannot detect '.tif' files,so it needs to be converted to jpg which results in high noise ratio when converted to binary file during detection,as it may remove many cell that are close to noise level or noise frequency.

3.1.1 Cellpose for Detection

As the problem shown in figure 3.2, detection is hard due to high noise and light interference in the data. So an algorithm was made using deep learning for detecting and classifying of different nuclei object for example, cells. So in this thesis, a part of cellpose [7] is use for detection of cell and remove all the light and noise problem.

A procedure for converting a manually annotated mask into a vector flow representation that neural networks can predict. A simulated diffusion process that begins at the centre of the mask is utilised to generate spatial gradients that point toward the cell's centre; it may also point indirectly around the corner. The gradient from each axis is merged to form a single normalised direction [7].

While cellpose is whole program but only a part is use which can clean the background noise and light and save it with clear cells and its elongation.The data of this thesis which posses cell with branches or elongation. Now, the predicted mask of

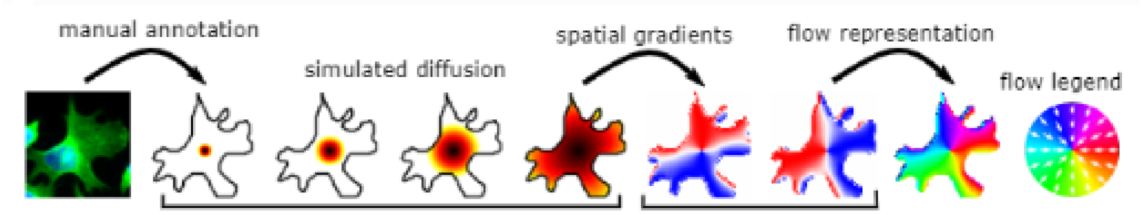


Figure 3.3: Detection and Transformation of cell in to vector flow that can be detected by machine learning algorithm from C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation,"Jan. 2021 `https://cellpose.readthedocs.io/en/latest/`

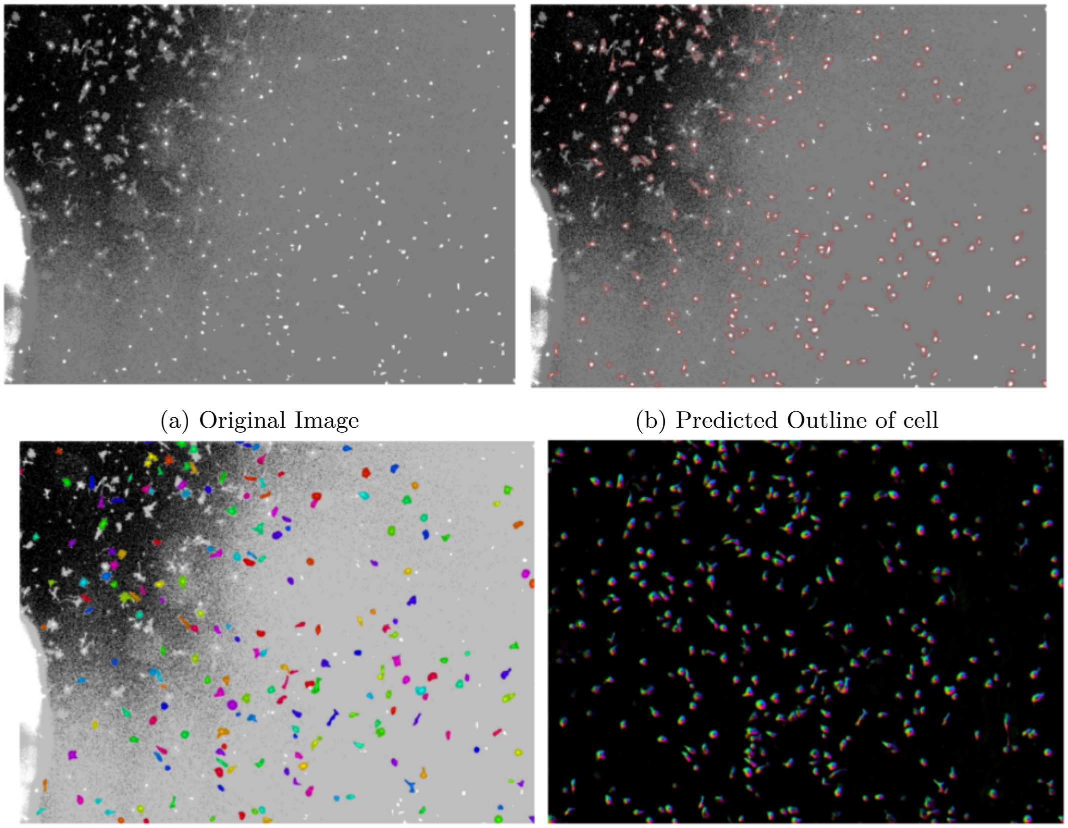
original image is use in cellpose.

- Model Design Their model is design based on watershed algorithm that uses the grayscale images to created a whole area map.

  These approaches operate best when the intensity profiles of the segmented objects decay smoothly from the centre, forming a single basin. Many types of cells, however, create multiple intensity basins, which is frequently attributable to the fluorescent marker's nuclear exclusion and inhomogeneous distribution along cell boundaries.These concerns prompted us to create an intermediate representation of an item that does constitute a single smooth area.A neural network was then trained to predict the topological maps' horizontal and vertical gradients, as well as a binary map that indicates whether a particular pixel is inside or outside the regions of interest (ROI). The horizontal and vertical gradients that produced vector fields were predicted by the neural network. All pixels belonging to a specific cell can be directed to its centre by following the vector fields using a technique known as gradient tracking. As a result of grouping pixels that converge to the same location, we may restore individual cells and their precise forms. Cell morphologies were improved further by deleting pixels indicated by the neural network to be outside of cells [7].

As shown in fig 3.4, as the process of detection in the cellpose and it will save the file.

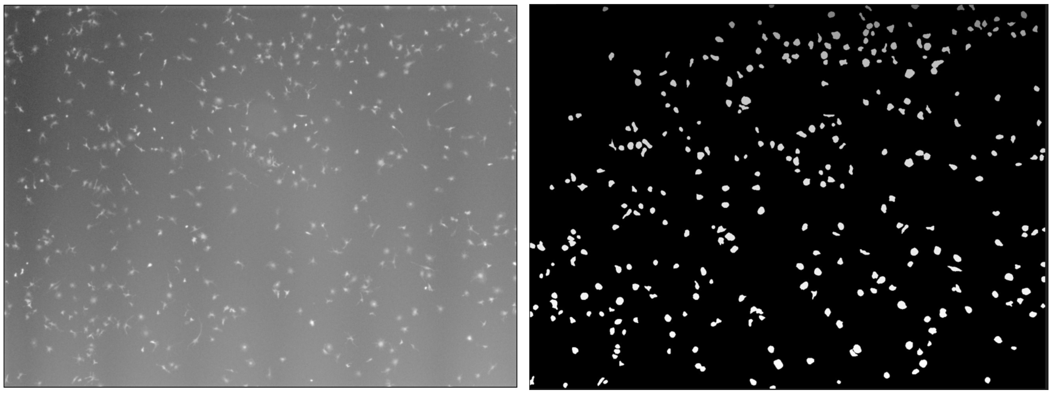
As shown in figure 3.5, the original image and the Segmented image using Cellpose, where image B is moe clear and sharp compared to original image which has lot of error. Comparing the Image B of fig 3.5 with Image A and B of fig 3.2. Use of cellpose is the right choice in case of this images.



(a) Original Image (b) Predicted Outline of cell

(c) Predicted Mask of cell (d) Predicted Cell Figure

Figure 3.4: Stages through which images are passed to get the clean error free images



(a) Original Image (b) Detected image using Cellpose[7]

Figure 3.5: Original image and Cellpose Segemented image

3.2 Features Extraction

There is no data regarding cell, so we have to extract many feature like area, centroid, convex_area, eccentricity, euler_number, extent, filled area, moments hu,major axis length, minor axis length,perimeter solidity, slice. This feature are also known as shape descriptors.
Shape features can be grouped into two classes: boundary features and region features.

The shape descriptors that are used in this thesis are define.

- Centroid: Distance of cell from x-axis(centroid-0) and y-axis(centroid-1).
- Area: Number of pixel in a shape.
- Convex Area: The portion of the convex hull that encloses the item is referred to as the object.
- Perimeter: the number of pixels within the object's border.Below mention formula uses 4 connected boundary that is distance measure using four corners as point in the image.

  $Perimeter = \sum_{i=1}^{N-1} d_i = \sum_{i=1}^{N-1} \mod (x_i - x_{i+1})$
- Major Axis length: the intersection of the (x,y) endpoints of the longest line that can be drawn through the item is known as Major Axis and the pixel distance between the major-axis endpoints of an item is called Major Axis length.

  $Major\_Axis\_Length = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$
- Minor Axis Length: the (x,y) ends of the longest line that can be drawn through the item while remaining perpendicular to the major axis is called minor axis and the pixel distance between the minor-axis endpoints of an item is called Minor Axis length.

  $MinorAxisLength = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$
- Eccentricity: the ratio of an object's short (minor) axis length to its long (major) axis length.The resulting number is a measure of object eccentricity ranging from 0 to 1.

  $eccentricity = \frac{axislength_{short}}{axislength_{long}}$
- Convexity: the degree to which an item varies from a convex object.This will be 1 for a convex object and less than 1 if the object is not convex, such as one with an uneven border.

  $eccentricity = \frac{convexperimeter}{perimeter}$

- Solidity: determines an object's density. A number of 1 denotes a solid item, while a value less than 1 denotes an object with an uneven border or including holes.

  $Solidity = \frac{area}{convexarea}$

- BBox_area(Bounding Box Area):A bounding rectangle of an item is a rectangle that completely encircles the object. The enclosing box has the same size as the main and minor axes.

  $BBox\_area = (Majoraxislength) * (minoraxislength)$

- Orientation: overall direction of the shape.

- Central Moments: μ_{pq} reflect attributes of a region that have been normalised in terms of location.Central moments are translation invariant, which means that two objects that are identical save for their centroids would have equal values of Central moments. However, central moments are not rotationally invariant. Central moments are not rotationally invariant; if an item is rotated, they will change.

  $\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q$ for p+q >1

- Normalised Central Moments:The normalised central moment is obtained by normalising the central moments with regard to the zeroth moment.The first central moment in x and y, η_{11} is the most typically normalised central moment. This offers a measure of the departure from the form of a circular area. A number close to zero describes a nearly circular area.

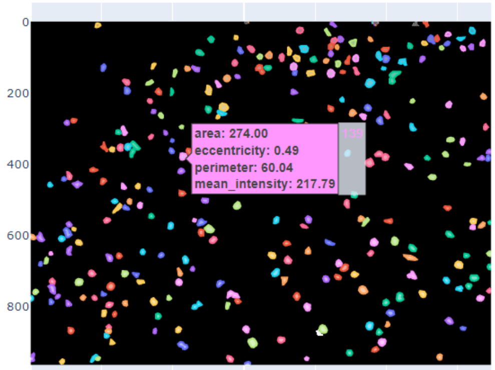
  $\eta_{pq} = \frac{\mu_{pq}}{\mu^{\gamma}{}_{00}}$

Above mention formula are described because this formulas and many more are used to calculate for few images in a cell.

As shown in fig 3.6, this image is use throughout the thesis for understanding and description, moreover this image is segmented and each cell has its own color for identification purpose.Also, if the cursor is hover over the cell, the cell information can be seen in it.

3.3 Cell Classification

The categorization of cells in machine learning is a difficult job in computer vision. This form of categorization is important in biomedicine. Several recent research have



area: 274.00
eccentricity: 0.49
perimeter: 60.04
mean_intensity: 217.79

Figure 3.6: Image with hover cursor to see the features of a particular cell

sought to train microscopic pictures to construct an Artificial Intelligence-based cell categorization system. These research produced the greatest findings and classified distinct types of cells based on their physical and biological characteristics [11].

In machine learning, there are two types of supervised learning problems: classification and regression [12]. Classification is the job of providing features to a machine learning algorithm and having the system classify the instances/data points into one of several discrete classes. Classes are categorical in nature; an instance cannot be classed as partially one class and partially another [12].

By using the above feature for each cell the csv file is created for hundreds of cell in each image. There arr 12 images for classification. In that 3 images have manual classification with their x and y axis provided. Then from manual data, euclidean distance is found and compared it with the centroid of the cell images saved as csv file.

Once identified and match the class was defined and 3 csv file are created for testing and training purposes, an example of csv file i shown in Table 3.1.

Training and Testing Data is split into 30:70 ratio for 3 images and it is put into different classifier to check which one has the best accuracy and prediction percentage.

The class are separated in 3 categories. Class A:no branches, class B: Few branches, Class C:Many branches.

Classifier used to check the classification are Random Forest, Ridge Classifier ,Gradient Boosting, Classifier, Nearest Neighbors, Linear SVM, Bagging Classifier, Decision Tree, Neural Net, Logistic Regression, AdaBoost, Naive Bayes, Gaussian Process.

3.3.1 Random Forest Classifier

A random forest is an estimator that employs averaging to increase predicted accuracy and control over-fitting by fitting a number of decision tree classifiers on different

area	46	56	52	84	58	75	73	47	22
bbox_area	72	168	77	143	130	120	168	88	96
centroid-0	3.673913043	7.589285714	7.25	10.53571429	12.36206897	16.53333333	19.23287671	19	20.09090909
centroid-1	32.67391304	850.375	1082.326923	189.0595238	570.1206897	84.37333333	165.4794521	224	1242.5
convex_area	52	76	60	96	73	88	107	58	42
eccentricity	0.902741527	0.964356514	0.769252987	0.775625039	0.876570993	0.719494848	0.662934524	0.860806471	0.966531158
euler_number	1	1	1	1	1	1	1	1	1
extent	0.638888889	0.333333333	0.675324675	0.587412587	0.446153846	0.625	0.43452381	0.534090909	0.229166667
filled_area	46	56	52	84	58	75	73	47	22
moments_hu-0	0.225055478	0.405828626	0.18313325	0.193253293	0.224599205	0.190971259	0.19689628	0.216387506	0.663880541
moments_hu-1	0.023952665	0.124409879	0.005921773	0.006911856	0.01963401	0.004447937	0.003074823	0.016219093	0.33859271
moments_hu-2	0.000194852	0.010898728	0.000651226	0.001577865	0.000555145	0.002127446	0.000480122	0.000907672	0.115345524
moments_hu-3	1.71E-05	0.006126172	8.49E-05	3.79E-05	1.02E-05	0.000330378	7.96E-05	0.000139781	0.037622185
moments_hu-4	-9.26E-10	4.93E-05	1.79E-08	-5.06E-09	-6.28E-10	2.73E-07	8.65E-09	4.73E-08	0.002105414
moments_hu-5	-2.64E-06	0.001971528	5.33E-06	-2.03E-06	-6.98E-07	2.20E-05	1.41E-06	8.20E-06	0.011702811
moments_hu-6	3.29E-10	8.66E-06	-8.72E-09	7.75E-09	4.48E-10	4.60E-08	1.29E-08	-1.55E-08	0.0013075
inertia_tensor-0-0	8.480623819	14.84151786	6.143121302	10.38931406	3.554399524	6.047288889	6.742728467	6.808510638	10.25
inertia_tensor-0-1	1.323724008	-9.243303571	1.447115385	2.650935374	2.785077289	2.239111111	1.974666917	2.446808511	-5.681818182
inertia_tensor-1-1	1.871928166	7.884885204	3.379807692	5.843962585	9.47235434	8.275555556	7.630699944	3.361702128	4.355371901
local_centroid-0	2.673913043	4.589285714	3.25	4.535714286	6.362068966	4.533333333	5.232876712	3	2.090909091
local_centroid-1	5.673913043	6.375	5.326923077	5.05952381	4.120689655	4.373333333	6.479452055	5	5.5
major_axis_length	11.82262587	18.43444655	10.40172598	13.62845041	13.00885724	12.4337633	12.13964007	11.36867989	14.80726585
minor_axis_length	5.08589709	4.877871381	6.646123722	8.6021954	6.260806269	8.635220979	9.088673909	5.78588989	3.798793029
moments-0-0	46	56	52	84	58	75	73	47	22
moments-0-1	261	357	277	425	239	328	473	235	121
moments-0-2	1871	3107	1795	3023	1191	1888	3557	1495	891
moments-0-3	15021	30951	12877	24995	6695	12436	29663	10711	7381
moments-1-0	123	257	169	381	369	340	382	141	46
moments-1-1	637	2156	825	1705	1359	1319	2331	590	378
moments-1-2	4249	21660	5185	11501	6273	7231	16475	3416	3434
moments-1-3	32407	234194	36765	91693	33177	46241	129117	22748	32364
moments-2-0	415	1621	725	2219	2897	2162	2556	581	192
moments-2-1	2039	15796	3325	8641	9713	7257	15207	2086	1798
moments-2-2	12791	171064	20285	53901	41411	36949	102887	11136	17430
moments-2-3	92369	1935742	140785	406765	203735	224031	762885	68776	171928
moments-3-0	1575	12029	3463	14733	25407	16042	19366	2763	988
moments-3-1	7453	128294	15045	50269	79215	46277	114471	8630	9744
moments-3-2	44425	1453482	89185	288401	315069	217801	756659	43136	97556
moments-3-3	305443	16922492	605205	2048941	1449189	1245395	5424609	249848	987162
moments_normalized-0-2	0.184361387	0.265027105	0.118136948	0.12368231	0.06128275	0.080630519	0.092366143	0.144861928	0.465909091
moments_normalized-0-3	-0.001517408	0.023309989	-0.004519049	0.0134398	0.003444868	0.004354507	0.00520454	0.002377157	0
moments_normalized-1-1	-0.028776609	0.165058992	-0.027829142	-0.031558754	-0.048018574	-0.029854815	-0.027050232	-0.052059756	0.258264463
moments_normalized-1-2	-0.004383358	0.034147845	0.007844163	0.000662305	0.001055557	0.00289288	-0.005936781	0.005348603	0.08633748
moments_normalized-1-3	-0.008282453	0.078096505	-0.007991243	-0.008817873	-0.007804491	-0.005567405	-0.006929577	-0.01902276	0.221003944
moments_normalized-2-0	0.040694091	0.140801522	0.064996302	0.069570983	0.163316454	0.110340741	0.104530136	0.071525577	0.19797145
moments_normalized-2-1	0.000694161	0.030299847	-0.002456263	-0.008753311	-0.006606341	-0.01386772	0.003385959	-0.008518146	0.096589055
moments_normalized-2-2	0.005755444	0.06083296	0.007576696	0.008307048	0.009560493	0.010018023	0.00637708	0.0138216	0.155381292
moments_normalized-2-3	0.00035234	0.024836905	-0.001664749	-0.001039616	-0.000206354	-0.002258191	0.001404815	-0.004092588	0.114461962
moments_normalized-3-0	0.000336675	0.022879849	-0.001827034	0.003326934	-0.000566417	0.012595074	0.003534128	0.004754314	0.081866978
moments_normalized-3-1	-0.002642837	0.051893406	-0.003916431	-0.004769767	-0.013501135	-0.010284725	-0.004083191	-0.008851603	0.121628397
moments_normalized-3-2	-7.35E-05	0.021721005	0.000951466	0.001067716	0.001122156	0.003128471	-0.000295356	0.003047313	0.105093533
moments_normalized-3-3	-0.000622666	0.028586364	-0.001046909	-0.001164466	-0.001782628	-0.00194185	-0.000700835	-0.003431987	0.130857149
orientation	-1.380284275	0.965356017	-1.16652917	-1.139759725	-0.377566303	-0.554543935	-0.674816697	-1.092217169	1.024656633
perimeter	27.10660172	35.97056275	28.14213562	37.21320344	32.97056275	37.17766953	32.93502884	29.3137085	21.62132034
perimeter_crofton	28.57653649	40.41332547	29.36193465	37.9618462	36.62108767	39.30260474	46.23643518	32.04345171	29.59197245
solidity	0.884615385	0.736842105	0.866666667	0.875	0.794520548	0.852272727	0.682242991	0.810344828	0.523809524
class	c	b	b	c	c	c	b	b	x

Table 3.1: Above mention Feature of first 9 cell of a single image

sub-samples of the dataset. If bootstrap is True (by default), the sub-sample size is controlled by the max samples argument; otherwise, the whole dataset is utilised to

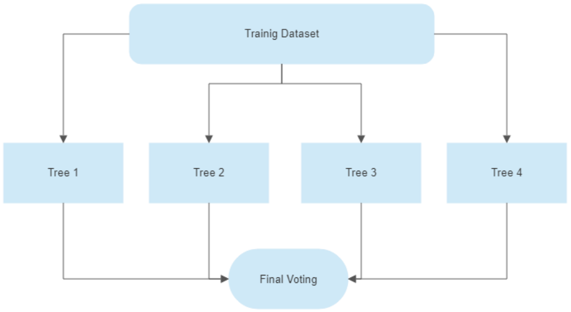


Figure 3.7: Random Forest Classifier

create each tree.

Random forest is a Supervised Machine Learning Algorithm which is frequently utilised in classification and regression problems. It constructs decision trees from many different samples and employs the majority vote for classification and the average of the regression [13]. It generates a new training subset from sample training data with replacement, and the final result is determined by majority vote is known as bagging.

As shown in fig 3.7, the training dataset has n number of records and from that m number of subset records are taken for sampling.Based on sample, a tree is created for single sample in this case tree 1,2,3.Based on this each tee generate an output and collective output are observe, and based on observation majority of voting are counted and decision is made on the resultant output of all trees.

3.3.2 Ridge Classifier

Based on the Ridge regression approach, the Ridge Classifier translates the label data to [-1, 1] and solves the issue using the regression method. The class with the greatest prediction value is chosen as the target class, and multi-output regression is used for multiclass data.

For binary classification, thee ridge classifier convert the label either into +1 or -1 based on the category. Then it checks on target class whether its greater than 0 then its positive otherwise its negative value

3.3.3 Gradient Boosting Classifier

Gradient boosting classifiers are a type of machine learning technique that combines many weak learning models to generate a powerful prediction model. When doing gradient boosting, decision trees are commonly employed. Gradient boosting models are effective because of its ability to categorise difficult datasets.

A loss function is used by the Gradient Boosting Classifier. Gradient boosting classifiers can employ an unique loss function and various standardised loss functions, but the loss function must be differentiable. Gradient boosting systems do not need to generate a new loss function each time the boosting technique is implemented; instead, any differentiable loss function may be applied to the system.Gradient boosting systems require two additional components: a weak learner and an additive component. Decision trees serve as the weak learners in gradient boosting systems. The additive component of a gradient boosting model stems from the fact that trees are added to the model over time, and when this happens, the existing trees are not changed; their values remain constant [12].

3.3.4 Nearest Neighbors

The k-nearest neighbours algorithm, often known as KNN or k-NN, is a non-parametric, supervised learning classifier that employs proximity to classify or predict the grouping of a single data point. While it may be used for either regression or classification issues, it is most commonly utilised as a classification technique, based on the idea that comparable points can be discovered nearby [14].

For classification, a class label is assigned on majority of vote it means the label will be use where there is more data of same time or of same similarity.So if there are two classes than majority of vote considered is more than 50%. But, for example, if there are 4 classes or label then majority of vote here will be more than 25% to considered the output to be valid.

3.3.5 Linear SVM(Support Vector Machine)

SVM is a linear model that can be used to address classification and regression problems. It can solve both linear and nonlinear problems and has a wide range of practical applications. This method draws a line on the plane to divide the data into classes is known as Linear Support Vector Machine. SVMs, to begin with, find a separation line (or hyperplane) between data from two classes. SVM is an algorithm that takes data as input and generates a line that, if possible, divides those classes. In an n-dimensional Euclidean space, a hyperplane is a flat, n-1 dimensional subset that separates the space into two unconnected sections [15].

3.3.6 Decision Tree

Decision Tree is a Supervised Machine Learning algorithm is a computer programme that uses a set of rules to make decisions in the same way that humans do. The idea

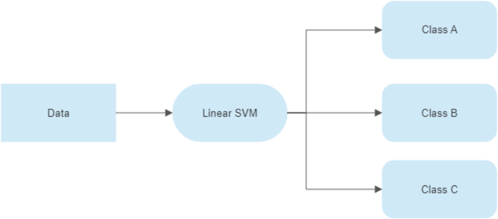


Figure 3.8: Support Vector Machine

behind Decision Trees is to utilise dataset characteristics to form yes/no questions and then partition the dataset until all data points belonging to each class are isolated. It is feasible to organise data in a tree form using this approach. Every time a question is added, a new node is added to the tree. The first node is known as the root node. The answer to a query divides the dataset and produces new nodes based on the value of a characteristic. If the process is terminated after a split, the final nodes formed are referred to as leaf nodes. Every time a question is answered, branches are formed, and the feature space is divided into discontinuous sections. One branch of the tree contains all data points that correlate to answering Yes to the inquiry suggested by the rule in the preceding node. The remaining data points are represented by a node on the other branch [16].

3.3.7 Neural Net

Neural networks are a collection of algorithms that are roughly based after the human brain and are meant to identify patterns. They analyse sensory data by categorising or grouping raw input using machine perception. They identify numerical patterns contained in vectors, into which all real-world data, whether pictures, music, text, or time series, must be transformed [17]. Neural networks aid in clustering and classification. Consider them a layer of grouping and classification on top of the data that is stored and managed. Also aid in the grouping of unlabeled data based on similarities between example inputs, and they classify data when they have a labelled dataset to train on.

3.3.8 Logistic Regression

Logistic Regression is a machine learning classification approach. The dependent variable is modelled using a logistic function. The dependent variable is dichotomous in nature, which means that there are only two potential classes (eg.: either the email is spam or not). As a result, while working with binary data, this strategy is applied.

The sigmoid function is used in logistic regression to transfer predicted values to probabilities. This function converts any real number between 0 and 1 into another value between 0 and 1. At each point, this function has a non-negative derivative and precisely one inflection point [18].

3.3.9 AdaBoost

It combines numerous weak classifiers to improve classifier accuracy. AdaBoost is a technique for iterative ensemble construction. The AdaBoost classifier creates a strong classifier by merging numerous low-performing classifiers, resulting in a high-accuracy strong classifier. Adaboost's main principle is to establish the weights of classifiers and train the data sample in each iteration to ensure accurate predictions of uncommon observations. Any machine learning method that takes weights on the training set can be used as the basic classifier.

At first, all observations are assigned identical weights. A model is constructed from a subset of data. Predictions are produced on the entire dataset using this model. The predicted and actual values are compared to compute the errors. Higher weights are assigned to data points that were mistakenly predicted in the following model. The error value can be used to calculate weights. For instance, the greater the mistake, the greater the weight ascribed to the observation. This approach is repeated until the error function remains constant or the maximum number of estimators is achieved [19].

3.3.10 Naive Bayes

A Naive Bayes classifier is a type of probabilistic machine learning model used for classification tasks. The classifier's core is based on the Bayes theorem [20].

$$P(A|\, B) = \frac{P(A|\, B)P(A)}{P(B)}$$

Using the Bayes theorem, we may calculate the likelihood of A occurring given that B has occurred. In this case, B represents the evidence and A represents the assumptions. In this situation, the traits are believed to be independent. In other words, the presence of one trait has no bearing on the presence of another. As a result, it is said to as naïve [20].

For example in this thesis, the cell classification is divided into 3 categories for classification, Class A, Class B and Class C.Also this classifier predict a cell category in either of this 3 classes, based on centroid or moments or moments hu as as shown in Table 3.1.

There are three types of naive bayes algorithm.

- Multinomial Naive Bayes:In this a category is based on number or frequency of words in a document.
- Bernoulli Naive Bayes:In this the catefory will answer in boolean form as yes/no, it check whether the fequency of words is greater or less than mention theshold.
- Gaussian Naive Bayes:In this the prediction values are continuous or not discrete.

3.3.11 Gaussian Process

Gaussian Processes are a generalisation of the Gaussian probability distribution that can be used to underpin advanced non-parametric machine learning methods for classification and regression. Gaussian probability distribution functions characterise random variable distributions, whereas Gaussian processes summarise function features. Gaussian processes necessitate the specification of a kernel that governs how samples relate to one another; particularly, it determines the data's covariance function. This is known as the latent function or "nuisance" function [21].

Chapter 4

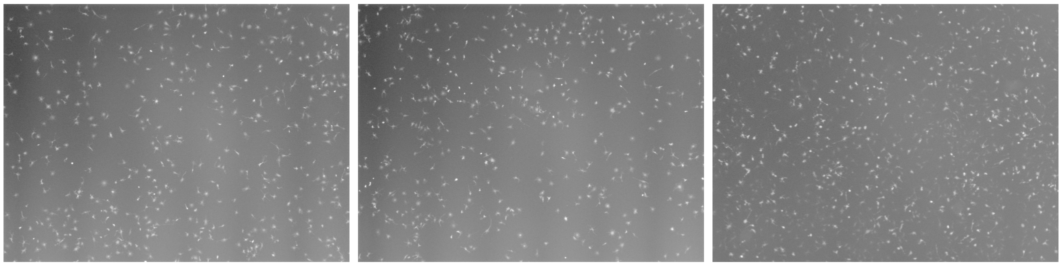
Data Description

4.1 Data Gathering

Most data can be categorised into four types in terms of Machine Learning: Diffeent Categories data, Data in numbers form, string or text, serial or timeline data.

The data for this thesis is gathered from Concordia university of Alberta, from Professor Matthew Churchward, he is a biology professor and he use to do this work of classfying and outlining the images manually.

He gave more than 888 different ".tif" extension images of cell for this thesis. From that, 12 images are chosen for classification, and 3 of the 12 images are already classified by him with imageJ. Where all 12 images were outlined but X and y axis co-ordinates with its correspondence classification(mainly A,B,C and x for error) is defined for 3 images. Because of this, the machine learning technique used is semi supervised learning.



(a) 04-18pA0A1B02x0y0w0 (b) 04-18pA0B3E08x1y0w0 (c) 07-12pA1A2B05x0y0w0

Figure 4.1: Three original Images for classificaion and Training & Testing purpose

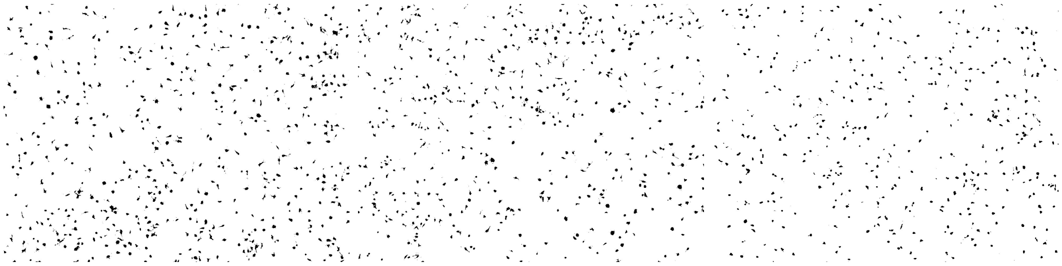
Fig 4.1 are 3 original image with ,Fig 4.2, as binary image of original image and Fig 4.3, where outline drawn on cell edges by manual labour.

04-18_pA0_A1_B02_x0_y0_w0.tif			
ROI	X	Y	Classification
1	635.3	17.5	A
2	757.9	17.4	A
3	87.6	12.8	A
4	310.8	12.8	A
5	1020.1	13.8	A
6	1235.2	13.6	C
7	1067.5	19.3	A
8	824.8	23.7	B

04-18_pA0_B3_E08_x1_y0_w0.tif			
ROI	X	Y	Classification
1	33.1	4.2	c
2	850.6	7.9	b
3	1082.8	7.7	b
4	189.6	11.0	c
5	570.6	12.8	c
6	84.9	17.0	c
7	166.0	19.7	b
8	224.5	19.5	b

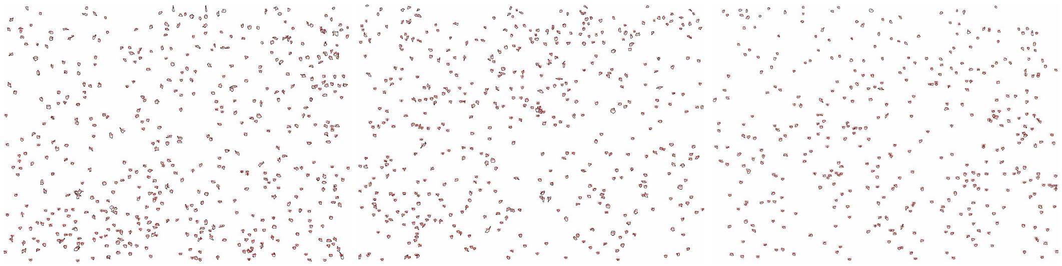
07-12_pA1_A2_B05_x0_y0_w0.tif			
ROI	X	Y	Classification
1	1059.3	9.1	c
2	338.2	10.3	c
3	470.6	10.7	c
4	208.6	14.9	b
5	592.1	22.2	c
6	126.6	23.3	b
7	101.7	21.1	c
8	913.8	23.6	b

Table 4.1: ROI(Region of interest) its X and Y co-ordinates of all 3 images but only first few cell for understanding with its classification, which is done by comparing the results by drawing outlines. 4.3



(a) 04-18pA0A1B02x0y0w0 (b) 04-18pA0B3E08x1y0w0 (c) 07-12pA1A2B05x0y0w0

Figure 4.2: Binary form of original image drawn manually



(a) 04-18pA0A1B02x0y0w0 (b) 04-18pA0B3E08x1y0w0 (c) 07-12pA1A2B05x0y0w0

Figure 4.3: Manual drawing of edges on cell on original image and classify them in A,B,C also done manually.

Chapter 5

Experimental Results and Discussions

5.1 Cell Segmentation

5.1.1 Otsu Thresholding

Otsu thresholding is based on pixel intensity.And we put input as grayscale image with thresholding(a value). While the output is based on thresholding and we get binary form of our input image. If the pixel intensity of an object in input image is greater than input threshold then it is marked as white and if threshold of object is less than that than it is marked black in background.

5.1.2 Multi-otsu threshold

As shown in fig 5.1, the threshold of the background where there is lot of light become white,which is not needed.So the use of multi otsu threshold in fig 5.2,to find the

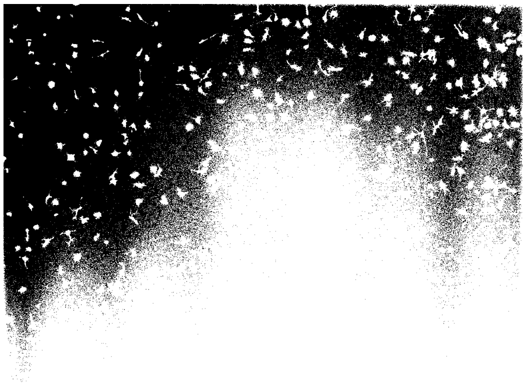


Figure 5.1: Classification of cell using otsu thresholding on original data

different region with different colour as to understand the threshold and adjust the threshold for cell, but it still ended in failure.Because of too much light interference, which cover the cell in half of the boundary and this was same for other 11 images where there is light and noise error in different pats of different images.

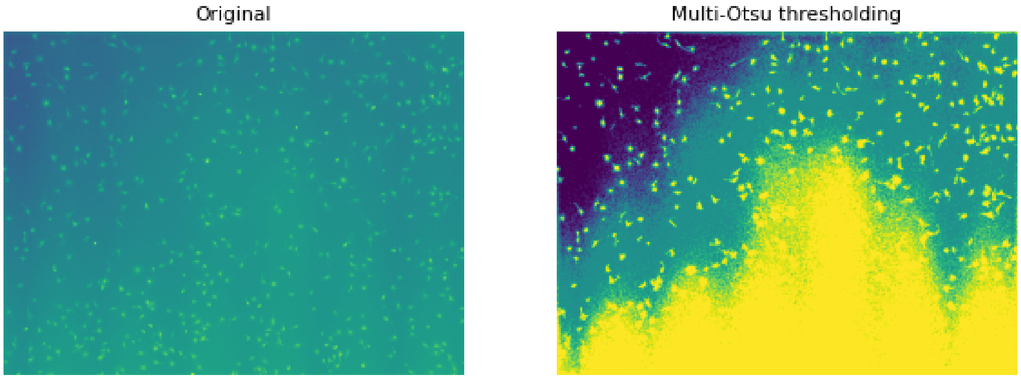


Figure 5.2: Multi otsu thresholding with original image

Also it can be seen that, the color of cell and the major yellow foreground which is an error have almost same threshold so it cannot be separated and otsu thresholding become a failure for cell segmentation for this particular data set.

5.1.3 Canny Edge Detection

Canny edge detection is a multi layer algorithm that use to find a structure of an image or edge print of a image.In this case, it the edges of each cell,the reason to use canny edge detection is because it gives very good structure of cell with high edges so it becomes easy to find the moments and moments-hu with centroid of each cell.Unfortunately, there is high noise error in background and lots of cell are close to noise error rate, because of this canny edge only detect those cell which are bright means a certain threshold,as shown in Fig 5.3, only few cell where detected.

5.1.4 Adaptive Thresholding

It is a form of detection which classify pixel as dark or light to form a grayscale image.It is very good where ther is illumination in an image as its show in this thesis images where the is lot of light illumination which needs to be removed.And in this case it was a success where the light illumination was completely disappear but the noise was high and it count the noise as cell, as shown in fig 5.4,the was use of different threshold in adaptive threshold with Gaussian blur and binary threshold but it was no able to eliminate the noise.Further if the illumination was canceled using this method,

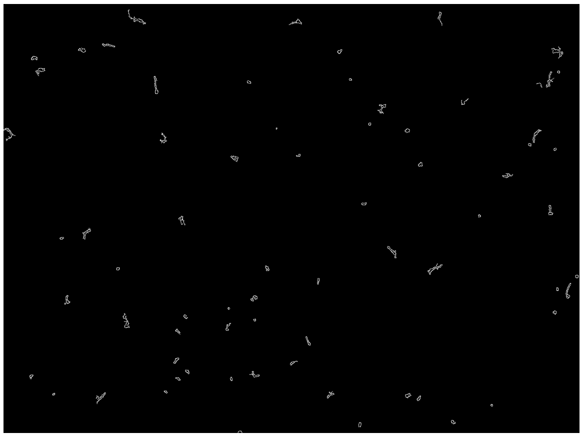
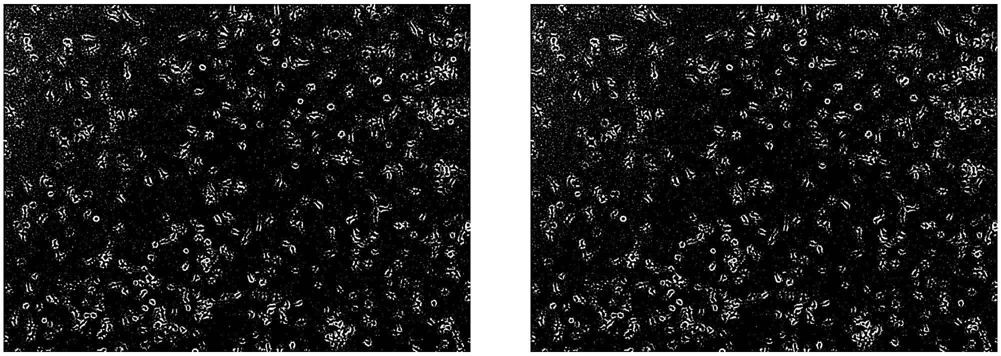


Figure 5.3: Use of Canny Edge Detector for cell segmentation

and put into otsu for cell noise elimination many cell with size of the noise will also disappear which was not expected result.



(a) Adaptive threshold with threshold 1

(b) Adaptive threshold with threshold 2

Figure 5.4: Adaptive threshold using different threshold and different result

5.1.5 Graph cut using slicing of image

Graph cut is an automated segmentation technique for separating foreground and background features in a picture. Good initialization is not required for graph cut segmentation. It can create scribbles on the image to indicate what is needed in the foreground and what needed in the background.in fig 5.5,the image is a part of the whole image which is taken to see whether it can detect the cell after segmenting the cell and separate in the foreground and background.

5.1.6 Contour

Contours are essentially a curve that connects all continuous points along the border that have the same hue or intensity. Contours are an effective tool for shape

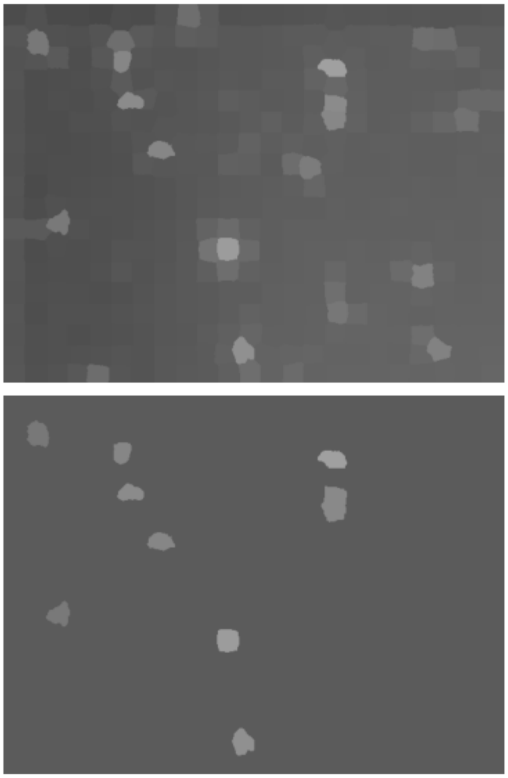


Figure 5.5: Image segmentation using graph cut slicing

analysis as well as object detection and recognition.As shown in fig 5.6, the contour detected not only cell but all the small error which can be seen as small dots and connect them along the border which makes the image look different, because before using contour there is need of using canny edge detection or some other detection method which in this case all this detection method failed.

5.1.7 Cell Segmentation with Single Layered Perceptrons

A feed-forward network based on a threshold transfer function is a single layer perceptron (SLP). SLP is the most basic type of artificial neural network, and it can only classify situations that are linearly separable with a binary target (1 , 0). In this case the image was first segmented as shown in fig 5.7, and then a part of the image or few cell were zoom in and clear the result but it was not effective for classification as region features extraction was hard as two close cell combined here after peceptron.

5.1.8 Watershed Segmentation

The watershed algorithm is built on gathering certain background and foreground information, and then utilising markers to run the watershed and discover the exact

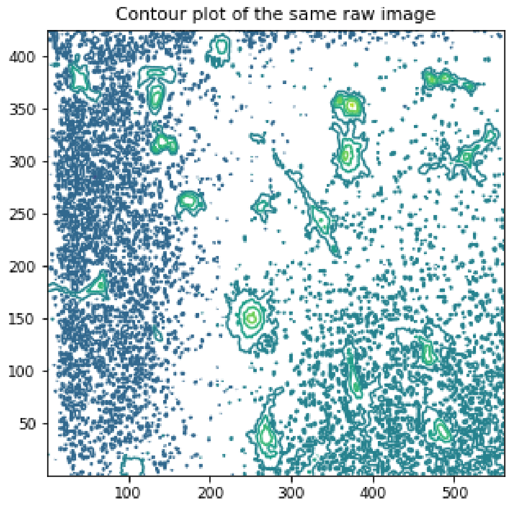
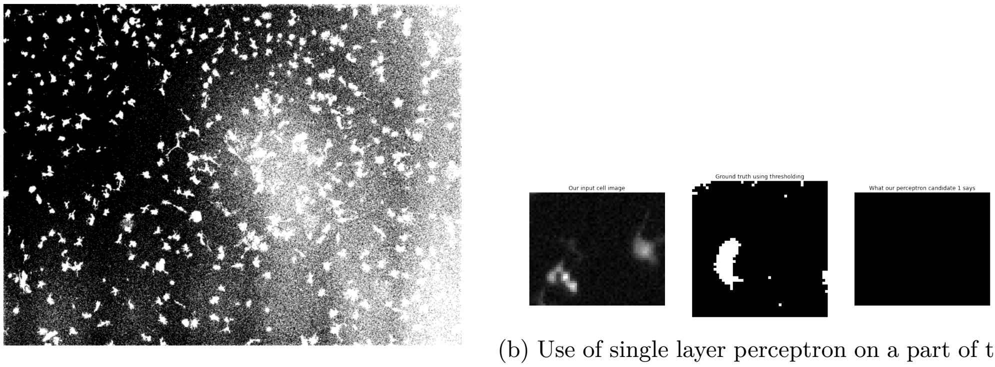


Figure 5.6: Image segmentation using graph cut slicing


Our input cell image
Ground truth using thresholding
What our perceptron candidate 1 says

(a) Image was segmented based on threshold

(b) Use of single layer perceptron on a part of the image

Figure 5.7: Cell segmentation with single layered perceptrons

borders. This technique is useful for detecting touching and overlapping objects in images.

For markers, it can be user defined, such as manually clicking and obtaining marker coordinates, or it can be defined by utilising certain defined algorithms, such as thresholding or any morphological operations. We can't use watershed algorithms directly because of the noise. The markers use before watershed is shown in fig 5.8.

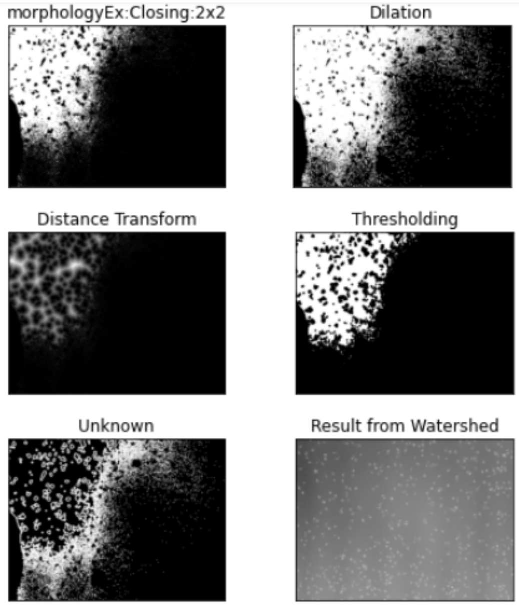


Figure 5.8: Image segmentation using graph cut slicing

5.1.9 Chan-vese Segmentation

The Chan-Vese segmentation algorithm is intended to segment objects with ambiguous borders. This algorithm is based on level sets that are iteratively evolved to minimise an energy defined by weighted values corresponding to the sum of differences in the sum of deviations in intensity from the average value outside the divided region from the average value inside the segmented region, and a term that is dependent on the length of the segmented region's boundary.

5.1.10 Cellpose

The need to use cellpose is because the failure of different segmentation and detection algorithm mention above and where cellpose comes into picture is because of its ability to remove light and noise interference and predicted the cell in an image as shown in figure 5.10. The use of watershed and u-net to form spatial gradient which give clear segmentation of the image.

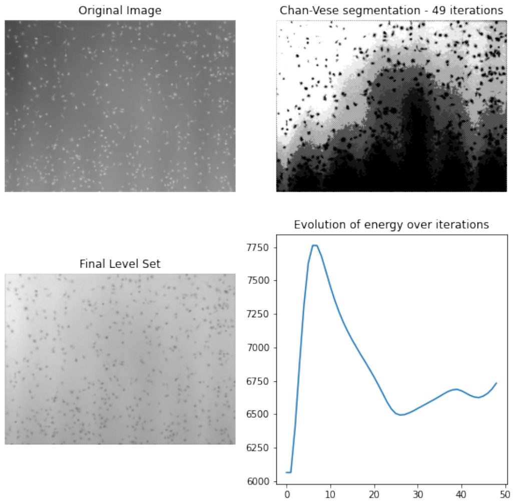


Figure 5.9: Image segmentation using Chanvese with 49 iteration and graph of energy change for 49 iteration

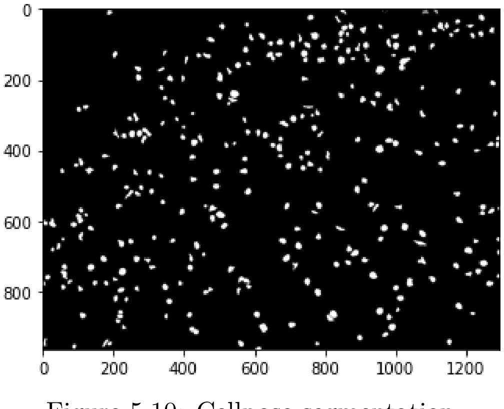


Figure 5.10: Cellpose segmentation

5.2 Cell Features

Table 3.1 as an example for showing how many feature of each cell was extracted from a single cell and how this feature are helpful in the classification of cell. But,first thing is after find the feature there is need to classify manually to each cell, so we use one of the 3 images and its x and y co-ordinates, from this co-ordinates euclidean distance of both manual and extracted feature distance was calculated using python algorithm and both distances were compared and put classification based on it manually for one csv file. Then this csv file was split into training and testing data.

5.3 Classification of cell

As mention in above in feature extraction, one file was created where classification done or data was created for training and testing for one time and it is split. After that it was put into different classifier and the classifier gave result, the maximum result got is with 60% accuracy.

	Training Accuracy	Test Accuracy
Random Forest	1.000000	0.595040
Ridge Classifier	0.613260	0.528930
Gradient Boosting Classifier	1.000000	0.520660
Nearest Neighbors	0.660220	0.512400
Linear SVM	0.488950	0.504130
Bagging Classifier	0.972380	0.504130
Decision Tree	1.000000	0.462810
Neural Net	0.450280	0.454550
Logistic Regression	0.400550	0.388430
AdaBoost	0.419890	0.355370
Naive Bayes	0.154700	0.140500
Gaussian Process	1.000000	0.016530

Table 5.1: Classification result of different classifier

As shown in Table 5.1,The training accuracy is very good for Random Forest, Gradient Boosting, Decision Tree and Gaussian Process. But, the best test accuracy we can get is 60% for Random Forest and 52% for two process Ridge Classifier and Gradient Boosting Classifier.Still there is room for improvement and we can increase the overall accuracy if this is perform on 12 images that were going to be used for classification. But this is the result for only first 12 images with one used for manual classification instead of 3 images.

5.4 Otsu Segmentation

The box plot,Fig 5.11, shown the score of each evaluation techique where PFOM value if negligible as it in power of -5 that is 10^{-5}

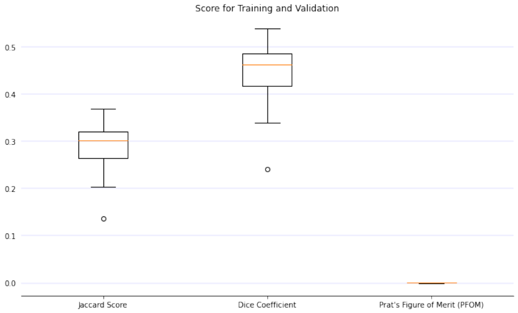


Figure 5.11: Box Plot of Jaccard Score ,Dice Coefficient and PFOM

- Jaccard Score : The Jaccard similarity index (also known as the Jaccard similarity coefficient) analyses members from two sets to determine which are shared and which are unique. It is a measure of resemblance between two sets of data that ranges from 0% to 100%. The larger the percentage, the closer the two populations are.
- Dice Coefficient : In this the segmented images and the gorundtruth images are compared at pixel level.
- Prat's Figure of Merit(PFOM) : An edge detection algorithm is used on an images and the quantitative value is found of an edge detected image.

	Jaccard Score	Dice Coefficient	PFOM
Image 1	0.13708907	0.241122835	4.66E-06
Image 2	0.369706266	0.539832919	2.41E-05
Image 3	0.296793212	0.457734061	2.84E-05
Image 4	0.204120711	0.339036957	3.29E-05
Image 5	0.306377705	0.469049194	3.47E-05
Image 6	0.28034931	0.437926288	2.09E-05
Image 7	0.31983602	0.484660238	2.44E-05
Image 8	0.281629474	0.439486575	4.62E-05
Image 9	0.218666267	0.358861606	3.40E-05
Image 10	0.305964369	0.46856465	4.61E-05
Image 11	0.32514748	0.490734028	1.72E-05
Image 12	0.347149522	0.515383804	1.66E-05

Table 5.2: Different Evaluation result on 12 classification images

Chapter 6

Conclusion and Future Works

The use for machine learning to automate cell detection in biology is a huge feat if it can save weeks or even months of time because research in biology is a very long process.As it takes time to experiment and take more time to conclude the best result.In this case, one of the professor problem was related to contrast phase image that is cell, where there is a need to count number of cell, with total number of cell in each categories and to make an observation of the specimen based on cell growth, the cell was divided into 3 categories which is based on its morphology, first categories was A which has lots of branch, second with few new branches and third is no branch a new cell.Now the need for this thesis is to save his weeks of time that he spent on one image to draw manually and concludes for one image which can be done in few hours even minutes with machine learning.

Where the first phase of machine learning is detection, how to detect cell where there is lot of image light interference and noise or error in it. Countless, detection algorithm where use like otsu detection algorithm, canny edge detection algorithm or watershed algorithm, but the problem was the right value of threshold that can detect cell yet the light interference as shown in many figure has almost same threshold as cell so it detect the light interference and it leads to bad result and many other cell has threshold value close to background,which eliminates the cell as background noise. So the need to use cellpose model which is made by stringer et al. which can remove bacground of varied threshold and predict the object in this case cell, and draw its morphology that can be understand by machine.Because their project was close to detecting and training nuclear level object example different kinds of cell.

The second phase was to extract cell feature like circularity, centroid, different kinds of moments, which is beneficial for training and testing for predicting cell in different image to define the cell categories.Ultimately eucildean distance was use for intial phase of detection and training of image. The third phase is to used the above extracted feature with manual drawing and categorizing of cell in machine learning classifier algorithm for testing, to check whether it is a success or not.Various classifiers where used to test, and from that Random Forest Classifer gave the best result with accuracy of 60%, this accuracy can be improved if 12 images which were initially selected for classification is used instead of only 2 from those 12 images.But overall the result was satisfactory as this was the first attempt on cell and its branches.

In the future as said, it accuracy can be increase but,by using 12 images instead of 2 images, instead of using machine learning algorithm there will be use of deep learning algorithm which are good for small objects like cell and for cell detection or segmentation will no use model made by other like cellpose instead will make a deep learning model for detection which will be give better result for detection for this thesis dataset.

Bibliography

[1] bruker.com, 2022. [Online]. Available: https://www.bruker.com/en/applications/academia-life-science/cell-biology/biophysics-and-biomechanics/cell-morphology.html#:~:text=In%5C%20bacteriology%5C%2C%5C%20cell%5C%20morphology%5C%20relates,cells%5C%20and%5C%20lymphoblast%5C%2Dlike%5C%20cells.

[2] Y. Li, J. Di, K. Wang, S. Wang, and J. Zhao, "Classification of cell morphology with quantitative phase microscopy and machine learning," *Opt. Express*, vol. 28, no. 16, pp. 23 916–23 927, Aug. 2020. DOI: 10.1364/OE.397029. [Online]. Available: http://opg.optica.org/oe/abstract.cfm?URI=oe-28-16-23916.

[3] D. Bhaskar, *Morphology based cell classification unsupervised machine learning approach*, 2017. [Online]. Available: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0345604.

[4] C. Di Ruberto and L. Putzu, "White blood cells identification and classification from leukemic blood image," 2013.

[5] Y. Qin, W. Wang, W. Liu, and N. Yuan, "Extended-maxima transform watershed segmentation algorithm for touching corn kernels," *Advances in Mechanical Engineering*, 2013. DOI: 10.1155/2013/268046. [Online]. Available: https://doi.org/10.1155/2013/268046.

[6] L. Nanni, M. Paci, F. L. Caetano dos Santos, H. Skottman, K. Juuti-Uusitalo, and J. Hyttinen, "Texture descriptors ensembles enable image-based classification of maturation of human stem cell-derived retinal pigmented epithelium," *PLOS ONE*, vol. 11, e0149399, Feb. 2016. DOI: 10.1371/journal.pone.0149399.

[7] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation," *Nature Methods*, vol. 18, no. 1, pp. 100–106, Jan. 2021. DOI: 10.1038/s41592-020-01018-x. [Online]. Available: https://doi.org/10.1038/s41592-020-01018-x.

[8] T. Vicar, J. Balvan, J. Jaros, *et al.*, "Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison," *BMC Bioinformatics*, vol. 20, no. 1, p. 360, Jun. 2019, ISSN: 1471-2105. DOI: 10.1186/s12859-019-2880-8. [Online]. Available: https://doi.org/10.1186/s12859-019-2880-8.

[9] F. A. Khan, U. Voß, M. P. Pound, and A. P. French, "Volumetric segmentation of cell cycle markers in confocal images using machine learning and deep learning," *Frontiers in Plant Science*, vol. 11, 2020, ISSN: 1664-462X. DOI: 10.3389/fpls.2020.01275. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2020.01275.

[10] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Processing Magazine*, vol. 29, pp. 140–145, Sep. 2012. DOI: 10.1109/MSP.2012.2204190.

[11] JENEEFAT, *Cell classification in machine learning*, 2021. [Online]. Available: https://www.madrasresearch.org/post/cell-classification-in-machine-learning.

[12] D. Nelson, *Gradient boosting classifiers in python with scikit-learn.* [Online]. Available: https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/.

[13] S. E. R, *Understanding random forest.* [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

[14] IBM, *K-nearest neighbors algorithm.* [Online]. Available: https://www.ibm.com/topics/knn#:~:text=The%5C%20k%5C%2Dnearest%5C%20neighbors%5C%20algorithm%5C%2C%5C%20also%5C%20known%5C%20as%5C%20KNN%5C%20or,of%5C%20an%5C%20individual%5C%20data%5C%20point..

[15] R. Pupale, *Support vector machines(svm) — an overview*, 2018. [Online]. Available: https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989#:~:text=SVM%5C%20or%5C%20Support%5C%20Vector%5C%20Machine,separates%5C%20the%5C%20data%5C%20into%5C%20classes..

[16] C. Bento, *Decision tree classifier explained in real-life: Picking a vacation destination*, 2021. [Online]. Available: https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575.

[17] C. Nicholson, *A beginner's guide to neural networks and deep learning*, 2021. [Online]. Available: http://wiki.pathmind.com/neural-network.

[18] A. Raj, *Perfect recipe for classification using logistic regression*, 2020. [Online]. Available: https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592#:~:text=Logistic%5C%20Regression%5C%20is%5C%20a%5C%20classification%5C%20technique%5C%20used%5C%20in%5C%20machine%5C%20learning,cancer%5C%20is%5C%20malignant%5C%20or%5C%20not).

[19] P. BANERJEE, *Adaboost classifier tutorial*, 2020. [Online]. Available: https://www.kaggle.com/code/prashant111/adaboost-classifier-tutorial/notebook.

[20] R. Gandhi, *Naive bayes classifier*, 2018. [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c#:~:text=A%5C%20Naive%5C%20Bayes%5C%20classifier%5C%20is,based%5C%20on%5C%20the%5C%20Bayes%5C%20theorem.

[21] J. Brownlee, *Gaussian processes for classification with python*, 2020. [Online]. Available: https://machinelearningmastery.com/gaussian-processes-for-classification-with-python/#:~:text=The%5C%20Gaussian%5C%20Processes%5C%20Classifier%5C%20is,algorithms%5C%20for%5C%20classification%5C%20and%5C%20regression..