

THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF PHYSICS

AUTOFOCUS OF PHASE-CONTRAST IMAGES OF YEAST CELLS USING MACHINE
LEARNING

YUJIE YAN
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Physics/Medical
with honors in Physics

Reviewed and approved* by the following:

Lu Bai
Associate Professor of Biochemistry & Molecular Biology and Physics
Thesis Supervisor

Richard Robinett
Professor of Physics
Associate Head for Undergraduate and Graduate Students
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Microscopic autofocusing is an essential technique for long-period image acquisition process. The hardware autofocus of the optical microscope detects small drifting distances of the sample slide. However, hardware autofocus fails especially when samples have unevenly coated coverslip, which makes the drifting distance exceed the limit of the optical detection system. My research aims to develop autofocusing software, to distinguish axial distances of phase-contrast yeast cell images (40-fold magnification). I first explored a deep Convolutional Neural Network, and then built other classification and regression by extracting features with different focus-measure methods. The classification models trained by focus-measure features can do a quick preliminary check of in-focus and out-of-focus images with over 99% accuracy. The shallow neural network with regression and selected combination of focus measure was able to distinguish different z-stacks taken from 0 micron to 17 microns with 1-micron step- size, both above and below focal plane. The RMSE of the best validation reached 0.33 μm and the best prediction accuracy was about 87% on the independent dataset, both of which outperformed the original method.

TABLE OF CONTENTS

LIST OF FIGURES	iii
LIST OF TABLES	iv
ACKNOWLEDGEMENTS	v
Chapter 1 Background	1
1.1 Hardware Autofocusing in Optical Microscope.....	2
1.2 Image Acquisition & TimeLapse	3
1.3 Focus Measures in Shallow Neural Network	5
1.4 Conclusion	6
Chapter 2 Convolutional Neural Network for Focus Determination.....	7
2.1 Introduction	7
2.2 Network & Training	7
2.3 Results & Evaluation.....	8
2.4 Conclusion	11
Chapter 3 Preliminary Training Using Classification.....	12
3.1 Introduction	12
3.2 Multi-Class Training	13
3.3 Binary-Class Training	16
3.4 Conclusion	18
Chapter 4 Feedforward Shallow Neural Network & Non-Linear Regression.....	19
4.1 Introduction	19
4.2 Methods.....	20
4.3 Results	22
4.4 Conclusion	24
Chapter 5 Feature Engineering	25
5.1 Introduction	25
5.2 Feature Importance in Classification.....	25
5.3 Individual Feature Evaluation in Regression	26
5.4 Collinearity Check & Implications	27
Chapter 6 Summary & Future Directions	29
Appendix A Table of Details of Thirty Focus Measures ^[2]	31

Appendix B Distributions of the Other Fifteen FM's.....	32
Appendix C Corresponding Classes for Multi- and Binary-Class Training with Classification	33
Appendix D Variance Plot of Principal Components	34
Appendix E Training Performance in Multi-class Classification with Different Combinations of FM.....	35
Appendix F Training Performance in Binary-class Classification with Different Combinations of FM.....	36
Bibliography	37

LIST OF FIGURES

Figure 1.1 Uneven Coating that Causes the Failure of Hardware Autofocus	1
Figure 1.2 Hardware Autofocus in Optical Microscope using a Detection Sensor ^[1]	2
Figure 1.3 Z-Stack Method for Image Acquisition	4
Figure 1.4 Example of Phase-Contrast Image of Yeast Cells at Different Z-Stack	4
Figure 1.5 Distribution of Fifteen FM (focus value vs. stack number).....	5
Figure 2.1 Training Process and Network Structure of CNN with Regression	8
Figure 2.2 CNN Training Progress of Overfitting	9
Figure 2.3 Edge Detection via Circular Hough Transform.....	10
Figure 2.4 Fraction of In-Focus Cells Captured by Hough Transform	10
Figure 3.1 PCA Plot for Observations labeled by 5-Class.....	15
Figure 3.2 Euclidean Distance Map of Observations labeled by 5-Class	15
Figure 3.3 Equation of Euclidean Distance	16
Figure 3.4 PCA Plot for In- and Out-of-Focus Clusters	17
Figure 4.1 Hypothesis Function of Non-Linear Regression Fitting.....	19
Figure 4.2 Equation of Mean Square Error ^[7]	19
Figure 4.3 Extracting Ratios of Focus Values using 3-Stack Method	21
Figure 4.4 Feedforward Shallow Neural Network Structure	22
Figure 4.5 Training Curve of Best Combo of FM	23
Figure 4.6 Comparison between Annotation and Prediction	24
Figure 5.1 Rank of Variable Importance in KNN.....	26
Figure 5.2 Individual Prediction Accuracies of Thirty FM's	26
Figure 5.3 Pearson Correlation Map for Thirty FM.....	28

LIST OF TABLES

Table 3-2 Comparison of Prediction Accuracy among KNN Multiclass Training	14
Table 3-3 Comparison of Prediction Accuracy among K-Means Binary-class Trainings.....	17
Table 4-1 Gradient Descent Algorithm ^[7]	20
Table 4-2 Comparison between focus value ratios predicts the direction.....	21

ACKNOWLEDGEMENTS

I would like to begin by thanking Dr. Bai for giving me the tremendous opportunities of exploring these wonderful research projects in the past two years in Bai Lab. It has truly been the best experience in my college life. I still remember the day that I had an interview with her, answering many questions about chemistry experiments and cell biology as a physics student. That's my first time contemplating my career path in interdisciplinary fields and it turned out to be the beginning. I would like to then thank my first mentor, Dr. Manyu Du, for treating me as a friend and student both at the same time, making me feel warm and comfortable as working in our lab or better phrased as a family. Dr. Bai and Dr. Du's kindness, enthusiasm, and accessibility were exactly what drove me to work with them in the lab. Then I would like to thank my current mentor, future Dr. Fan Zou, for putting up with me as a programming neophyte who liked to ask thousands of questions at a time. I appreciate his many-minute-long explanations on each of my questions, intensive hands-on guidance, and his sense of organization, allowing me to know how to be a good researcher.

I then would like to thank Dr. Richard Robinett for his unwavering supports on each of my decisions and truly sincere advice on each of my questions, both on life and academics. I still remember every advising meeting and meaningful talks that we had during the past four years. The help and care offered by Professor Robinett made my college experience much better than what I can expect at Penn State.

I would also like to thank Penn State Physics REU and NSF for the support of research funding during my REU in 2019.

I also appreciate the help from other lab members – Hengye, James, Yi, and Hungyo. Thank them for spending time chatting with me about my concerns and worries during my first REU summer, every day when I struggled with results and the final presentations. I learned that the true meaning of research is about exploration and dealing with unknowns all the time through talking with them after lunch. I hope to have more time working in the lab (if there's no pandemic) and maybe have lunches with Jenna and Holly, who I wish I could know better.

For my college friends outside the lab, Elizabeth, Hugo, Hinkal, Waka, Brenda, and Summer, I would like to thank them all for their encouragements and accompaniment during the pandemic. Many of them had their thesis project going on and the mutual inspirations drive through this road to completion.

Finally, I would like to give a virtual hug to my parents and my grandparents, who were always there whenever I needed them, telling me to take some time off and relax rather than spending all my time on academics. I always remember their enthusiasm and excitement on my publication and presentations of research. I am truly grateful to have them as my family. I hope to go back home this year and meet them in person instead of just video calls.

Chapter 1 Background

Microscopic autofocus is an essential technique for the long-period image acquisition process (i.e. movies) in a biophysics lab to explore chromosomal dynamics in yeast cells. The hardware autofocus of the optical microscope detects small drifting distances of the sample slide, but it sometimes fails especially when the sample is covered with uneven coating. Before a continuous live-cell imaging is initialized, the focal position is manually preset based on the focus of cells at one position. During this process, small drift can be detected and adjusted by the hardware autofocus by moving the sample stage, but the uneven coating always causes large drifts that exceed the limit of the hardware detecting system (see Figure 1.1). Even though these large drifts can be manually adjusted once we spot them images, it is not efficient at all to stare at the screen, particularly for movies that last hours. A gradient-based software autofocus method using machine learning has been applied to compensate for the large-distance limitation of the hardware detection sensor. However, the implementation of this software is not perfectly reliable and when it fails, the whole movie can go into waste.

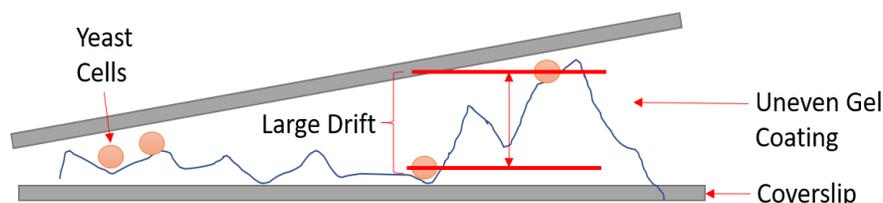


Figure 1.1 Uneven Coating that Causes the Failure of Hardware Autofocus

Thence our aim of this project is to explore other software autofocus methods, through unsupervised and supervised learning, ensuring that samples (i.e. yeast cells) are always at focus

position during long live-cell imaging processes. Methods involve Convolutional Neural Network (CNN) which has no feature specification or output labeling (i.e. deep structure and unsupervised learning algorithm), and classification/regression using focus measure as predictors with output labeled (i.e. shallow structure and supervised learning algorithm). Before I discuss these methods, I would like to introduce the microscope, TimeLapse program, and focus measures (FM) in this chapter.

1.1 Hardware Autofocusing in Optical Microscope

Figure 1.2 shows the imaging of the Leica optical microscope and how autofocusing adjustments by the detection sensor works. A LED light ray passes the objective and gets reflected by the sample stage, pointing to the detection sensor. The focal position corresponding to certain

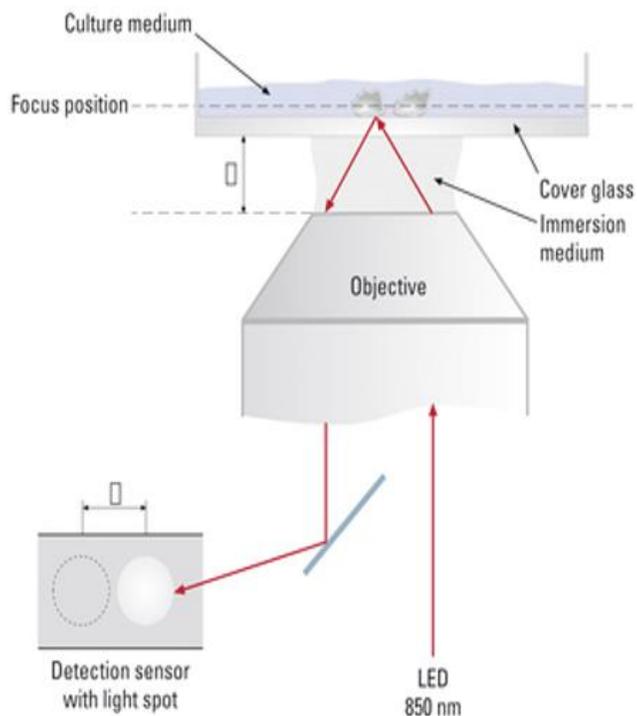


Figure 1.2 Hardware Autofocus in Optical Microscope using a Detection Sensor ^[1]

light spots on the detection sensor needs to be pre-set so that the sensor can tell how far the focus of each position differs from others vertically if the light spot moves. However, as I mentioned before, such detection of focus differences by the sensor has a limit and if they are out of range, hardware autofocus will not work well. As a result, we started exploring software autofocus to improve image quality, mainly through machine learning, which requires four major steps: preprocessing, network construction, network training, and refinement.

1.2 Image Acquisition & TimeLapse

The in- and out-of-focus images for training are taken using a z-stack method implemented in the TimeLapse program. The camera takes images at different z-positions with preset step size and these images are called z-stacks. Take the example shown in Figure 1.3. Starting from the most top z-stack, the camera moves down with a step size of 0.3-0.4 microns and takes 9 consecutive images according to the size of sphere-shaped hypoid yeast cells. The stacks taken above the focal plane are labeled with a minus sign. “- 1 um” means 1 micron away above the focus. The middle stack, which is the 5th, is expected to be exactly in-focus if the stage doesn't shift during imaging. However, this isn't always consistent so the focal/central stack must be selected by human eyes.

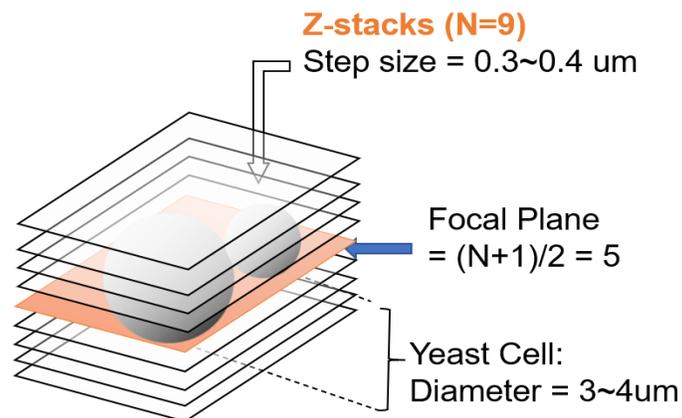


Figure 1.3 Z-Stack Method for Image Acquisition

By examining the visual features of phase-contrast images of yeast cells taken at different z-positions, we found that the in-focus cells tend to have a sharp edge and a bright “halo” around the edge while the out-of-focus cells have multiple bright “halos” (see Figure 1.4).

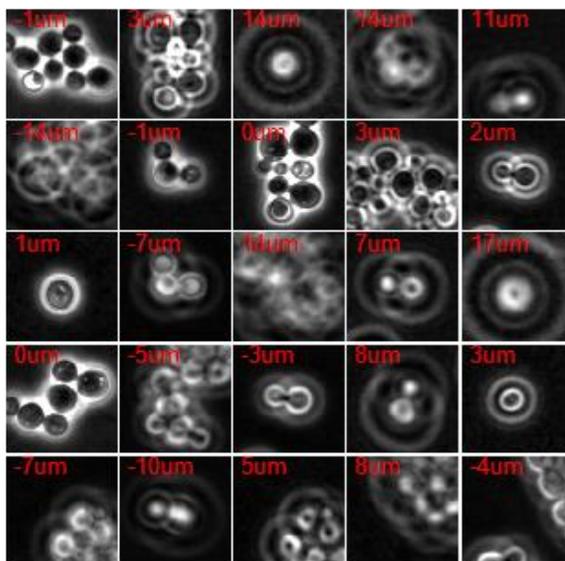


Figure 1.4 Example of Phase-Contrast Image of Yeast Cells at Different Z-Stack

1.3 Focus Measures in Shallow Neural Network

In supervised learning, certain features and patterns around the cell edges can be explored and exploited in the determination of focal positions. We defined several focus measures, including contrast, curvature, gradient, gray level variance, and wavelet transform, etc., which can be extracted from the image data and feed into a shallow neural network to determine the focal point. Consequently, thirty focus measures were found to extract focus values, whose distributions are demonstrated in Figure 1.5 (also see Appendix A for full names and Appendix B for the other fifteen features).^[2]

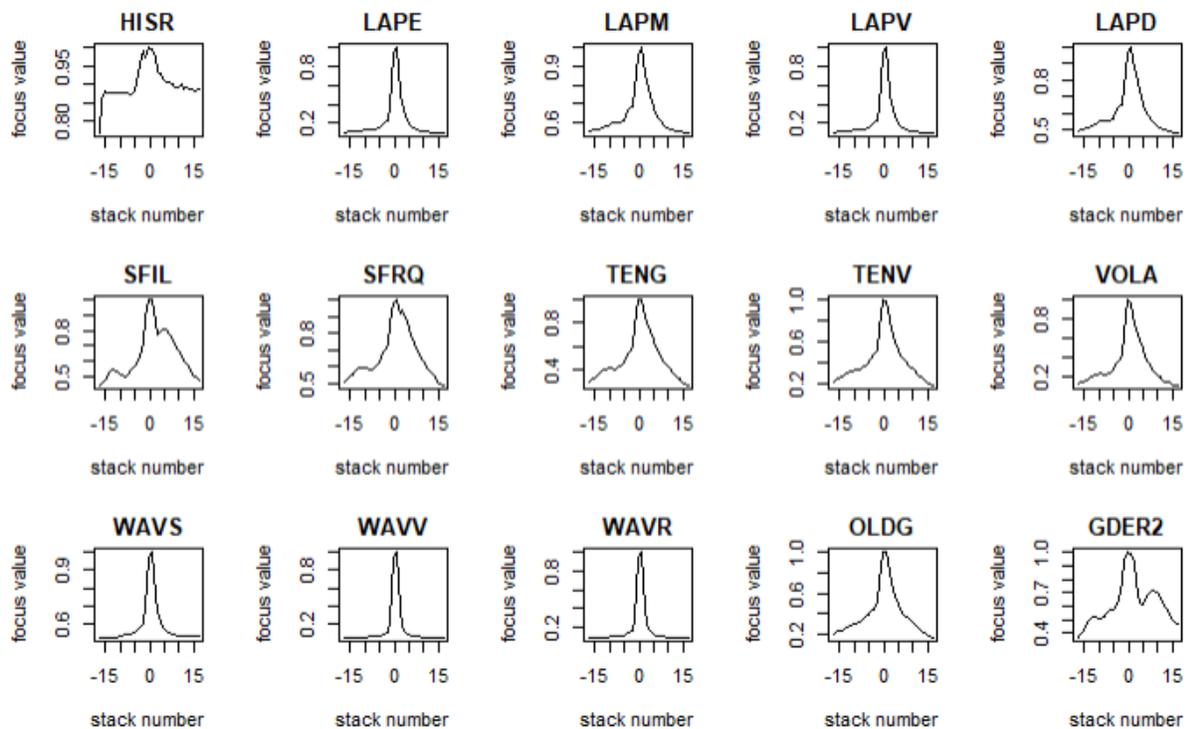


Figure 1.5 Distribution of Fifteen FM (focus value vs. stack number)

After normalization in terms of the central, maximum focus value, most of their distributions should look like a “bell-shaped” curve. Even though these focus measures are

featured by their Gaussian distributions, which a traditional nonlinear regression (e.g. curve fitting) might fit well, we still believed that the neural network could be better at coping with nonlinearity.^[3] Finally, since focus measures were incorporated as predictors, such nonlinear regression with a neural network didn't require as many hidden layers as CNN's, whose deep structure was essential for feature extraction.

1.4 Conclusion

In this chapter, I introduced the objectives of this autofocus project, the apparatus, the preparation of z-stack images, and focus measures used to extract features from images. The software autofocus will mainly involve preprocessing of images and different models in machine learning and their training with processed images. In the following chapters, I will discuss the machine-learning models and their training, including Convolutional Neural Network, supervised and unsupervised classification, and shallow neural network with non-linear regression.

Chapter 2 Convolutional Neural Network for Focus Determination

2.1 Introduction

Artificial Neural Networks (ANN) with multiple layers have been considered a powerful tool in handling large data using deep learning techniques over the last few decades. Especially, Convolutional Neural Network (CNN), as the most popular deep neural network, has been widely and successfully used in image processing, such as pattern recognition, image classification, and natural language processing.^[3] MathWorks released a document of training CNN for regression, stating that the prediction of continuous data, such as distances and angle, can be achieved by including a regression layer at the end of CNN.^[4] To predict the axial distances between the position where images were taken and the focal plane, I trained this model with 9-stack phase-contrast image data sets.

2.2 Network & Training

To collect images for training and testing, we took in-focus ($z=0$) and out-of-focus ($z= \pm 0.1\sim 2$ micron with a step of 0.3 micron) images of hypoid budding yeast both above and below the focal plane. More than 4000 images were randomly distributed for training and the remaining 1000 images were used for testing. The data set contains both the downsized, normalized phase-contrast images, and the corresponding z positions.

The image preprocessing involved cropping a certain size of images from the original images with a larger field of view and much more pixels. This helped reduce time cost and remove

as much noise as possible from the background or dead cells. 512x512 (pixel) image data are used in the following training mentioned in this thesis.

After cropped images were imported in training, convolutional filters walked through each image to extract features and the regression layer at the end of the network helped to classify images with continuous values of distances according to patterns computed and learned by middle activation layers (see Figure 2.1).

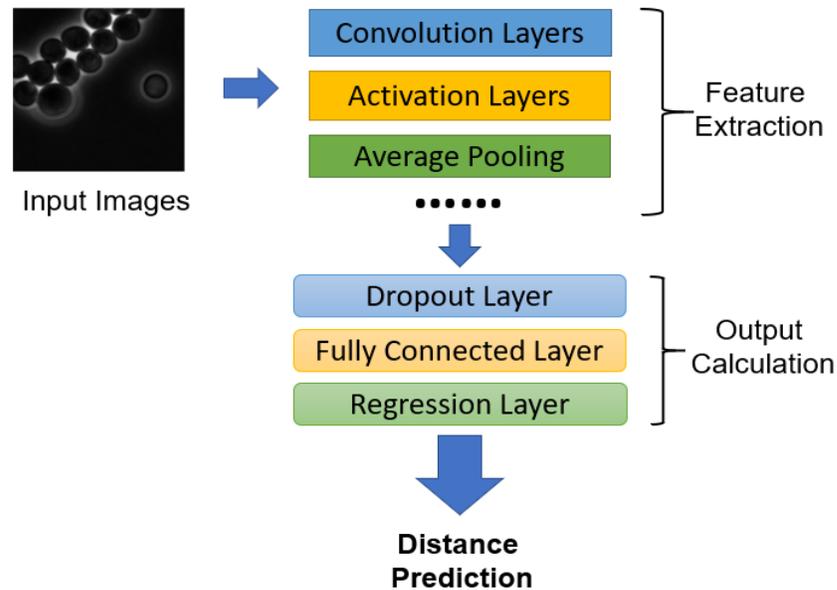


Figure 2.1 Training Process and Network Structure of CNN with Regression

2.3 Results & Evaluation

After the first round of training, we found that the model was overfitted based on the large root-mean-square-error (RMSE). A good fitting would demonstrate a monotonical decrease of

both RMSE and loss function, with similar validation training curves but we didn't observe it in the training curve, shown in Figure 2.2.

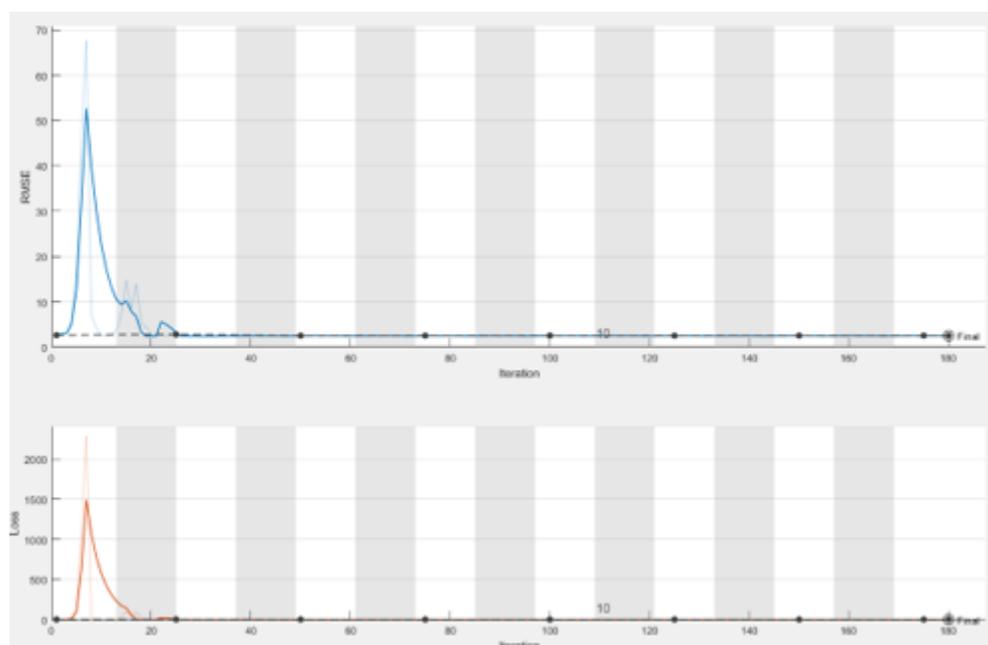


Figure 2.2 CNN Training Progress of Overfitting

There are few reasons that can explain this overfitted training, including the difficulty of learning assignments and the processing of original images. First, the step size of z-stacks may be too small for the model to tell the differences. We did an explicit feature analysis through Circular Hough Transform, which returns the relative intensity and sharpness of cell edges (i.e. metric value). Figure 2.3 shows that out-of-focus images tend to have more than one circle detected for one cell due to the haloes around the edge. By examining a lot of images and cells, we set a cutoff (metric value > 1) of in-focus cells metric value and check how different these cell edges are by defining a fraction of in-focus cells they selected. Figure 2.4 shows the fraction drop at around 1 micron, suggesting that we should consider 1 micron as the step size. Second, CNN may be more sensitive to the orientation, shape, and number of cells on an image based on its existing

applications. Without further control, it cannot prioritize features on cell edges, in which we are interested.

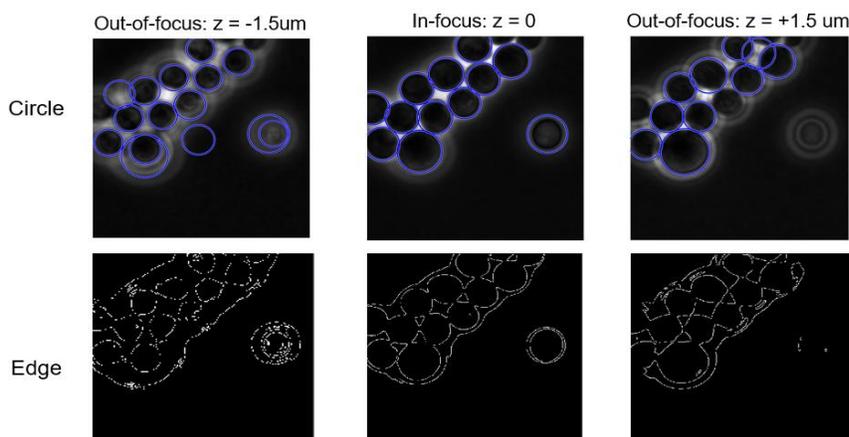
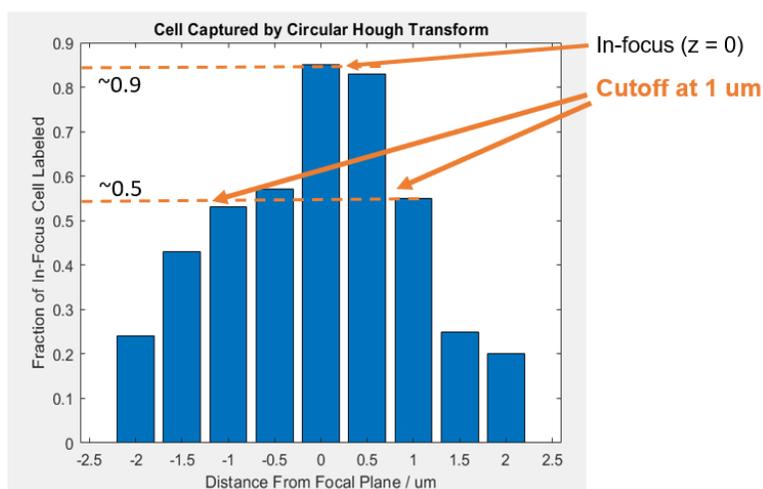


Figure 2.3 Edge Detection via Circular Hough Transform



$$\text{Fraction of In-focus Cell Captured} = \frac{\text{Number of Cells (Metric} > 1)}{\text{Total Number of Cells}}$$

Figure 2.4 Fraction of In-Focus Cells Captured by Hough Transform

Finally, 100x-magnification images may capture too many useless details. Wei and Roberts have used CNN and classification layer for detection of focal positions of yeast cell images, and they achieved this using 10x-magnification images which were taken with 5-micron step size and

a much larger range (e.g. 0 to 75 microns).^[5] Consequently, we decided to switch to 40x-magnification images with 1-micron step size and 35 stacks (ranging from 0 to 17 microns), which is also more commonly used in long-time image acquisition (i.e. movies).

2.4 Conclusion

CNN demonstrates its potential at recognizing focal position as the literature shows but it is not suitable for our specific task. Moreover, as a deep neural network, the training is usually time-consuming and requires much more data than regular machine learning methods. As a result, besides the adjustment of data, including switching to 40x magnification, using larger step size at 1 micron, and taking 35 stacks at one position, we also wanted to try traditional classification and shallow neural network with focus measures as pre-controlled parameters.

Chapter 3 Preliminary Training Using Classification

3.1 Introduction

Before we shift to regression, which may indeed be a more straightforward way to fulfill our purposes of predicting continuous distances, it is necessary to do some preliminary training to check these FM's. This aims to roughly evaluate how well FM's can distinguish in- and out-of-focus images using classification models. Here then comes the problem of how to separate classes. To figure this out, I look through the distributions of different focus measures and find the common pattern of “suitable” ones from our perspective. Focus values of one set of images are all normalized based on the maximum in this set, so the values range from 0 to 1.

As it was mentioned in Chapter 1, most FM distributions are “bell-shaped”, where the maximum is always in the middle, the curve is symmetric concerning the maximum, and focus values decrease monotonically as the z-stack goes further from the focal plane (i.e. $z=0$). It is also interesting that within a certain distance (e.g. less than 4 or 5 microns), the focus values decrease so drastically that the model can probably easily distinguish differences. However, starting from 4- or 5-micron separation from the focal position, the focus value (e.g. intensity, contrast, gradient, blurriness, etc. See Appendix A for full names of all thirty features) barely changes as we move to further stacks. According to the patterns of distributions, z-stacks were separated into multi classes, including 5-class, 6-class, and 7-class, to test the potential of classification models. I also set up an assignment of binary classes, with only in- and out-of-focus, to check if FM's can perfectly select out very “bad” images, meaning very far from the focal position. Classes are shown in Figure

3.1 where Arabic numerals represent distances from the focus and Roman numerals represent class numbers (see more details in Appendix C).

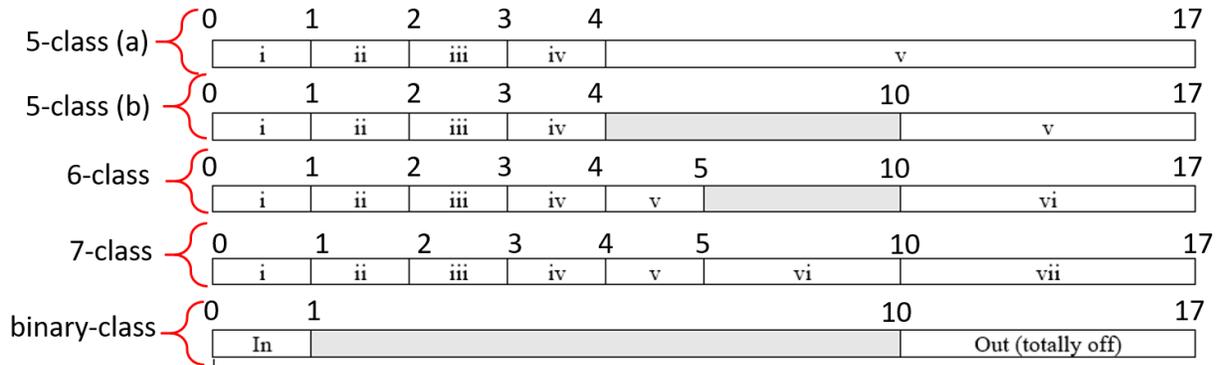


Figure 3.1 Demonstration of corresponding Multiclass Separation

3.2 Multi-Class Training

For the multi-class training, I selected one commonly used classification model, K-Nearest Neighbors (KNN). In KNN, data points for training are plotted in a multi-dimensional space, which depends on the number of parameters. Then the test data are also plotted in the same space to be compared with the existing, classified training data. “Given a positive integer K and a test observation x_0 ”, the KNN model first identifies K closest data points among training observations, namely, “neighbors” and assign x_0 to the class where most neighbors lie. ^[6]

During data pre-processing, FM mathematical functions were applied to extract focus values from raw images and the focus values were normalized within one set of 35 z-stack (from -17 micron to +17 micron) based on the maximum focus value. During training, I also used 10-fold cross validation to test my training results, which were evaluated by prediction accuracy. The

prediction accuracy is defined by the ratio of the number of test observations classified into the correct category to the total number of test observations.

It was found that with larger number of classes, the prediction accuracy decreased as expected. Moreover, the prediction accuracy of multiple FM combinations tends to be higher than the one trained by the “original gradient-based method” (OLDG). The best FM combination was accomplished by feature selection, which would be discussed in detail in Chapter 5.

Prediction Accuracy	5-class (> 4um)	5-class (>10um)	6-class	7-class
Only OLDG	0.89 ± 0.007	0.802 ± 0.005	0.744 ± 0.007	0.668 ± 0.011
Best FM Combo	0.925 ± 0.005	0.926 ± 0.005	0.901 ± 0.007	0.825 ± 0.011
All 30 FM	0.906 ± 0.004	0.902 ± 0.008	0.805 ± 0.010	0.745 ± 0.014

Table 3-1 Comparison of Prediction Accuracy among KNN Multiclass Training

Moreover, I plotted part of the observations, with 5-class labels, on a PCA plane to visualize the separations among different classes. PCA stands for Principal Component Analysis, “a popular approach for deriving a low-dimensional set of features from a large set of variables”.^[6] In Figure 3.2, the first two principal components could capture almost 90% of the characteristics (also see Appendix B).

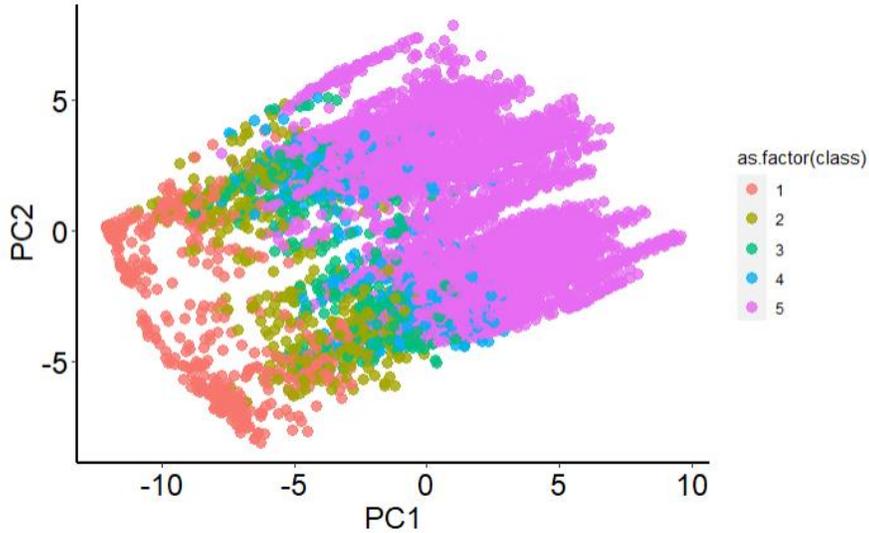


Figure 3.2 PCA Plot for Observations labeled by 5-Class

The PCA plot below didn't demonstrate clear separations among classes. Instead, the differences may not be large enough to distinguish small distances since there were overlapping regions all over the classes. The data points looked very continuously connected and regression methods were thought to fit better. A distance map shown below (Figure 3.3) also supported the ambiguous separation among labeled classes.

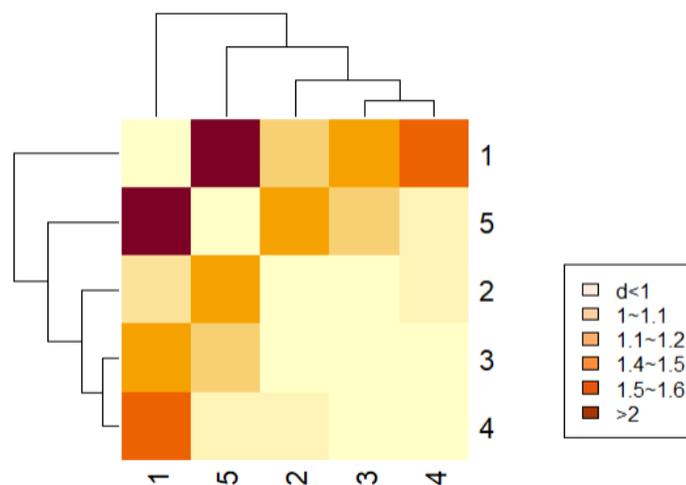


Figure 3.3 Euclidean Distance Map of Observations labeled by 5-Class

The Euclidean distances were calculated based on the following equation in 30-dimensional space where a_i and b_i represent two data points and the subscript i ranging from 1 to 30 represents thirty dimensions/predictors/FM's.

$$\text{Euclidean Distance} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \dots + (a_{30} - b_{30})^2}$$

Figure 3.4 Equation of Euclidean Distance

Compared with other average inter-class distances, the one between Class 1 ($0 \leq d \leq 1$) and Class 5 ($d > 10$) is relatively large and may be adequate to distinguish differences. To confirm this thought, I decided to do a binary classification for only in- and out-of-focus categories.

3.3 Binary-Class Training

For binary-class training, I selected K-means clustering to classify in- and out-of-focus (see Table 3-1 for definition) images. Each training observation was randomly assigned to one of the K clusters and within one cluster, the K-means algorithm would minimize the Euclidean distances among data points by reassigning them into the ideal cluster.^[6] I used both the original FM parameters and PCA parameters (i.e. dimension reduced) and compared their performance based on prediction accuracy under three conditions, only OLDG, best FM combination, and all 30 FM. Note that PCA must have more than two predictors. It was found that the best prediction performance using K-means and multiple FM combination was better than using only OLDG and all 30 FM's. The overall prediction accuracy was almost perfect, especially ones with multiple, selected FM combinations (see Table 3-3).

Independent Prediction Accuracy	PCA + K-Means	K-Means
Only OLDG	×	0.9978
Best FM Combo	0.9992	0.9993
All 30 FM	0.9949	0.9975

Table 3-2 Comparison of Prediction Accuracy among K-Means Binary-class Trainings

The visualization of in- and out-of-focus clusters can also explain this high prediction accuracy (see Figure 3.5). PC1, the first principal component, capturing almost 90% of the characteristics of focus measures, successfully distinguishes these two classes with a relatively large separation, namely, large average inter-class Euclidean distance.

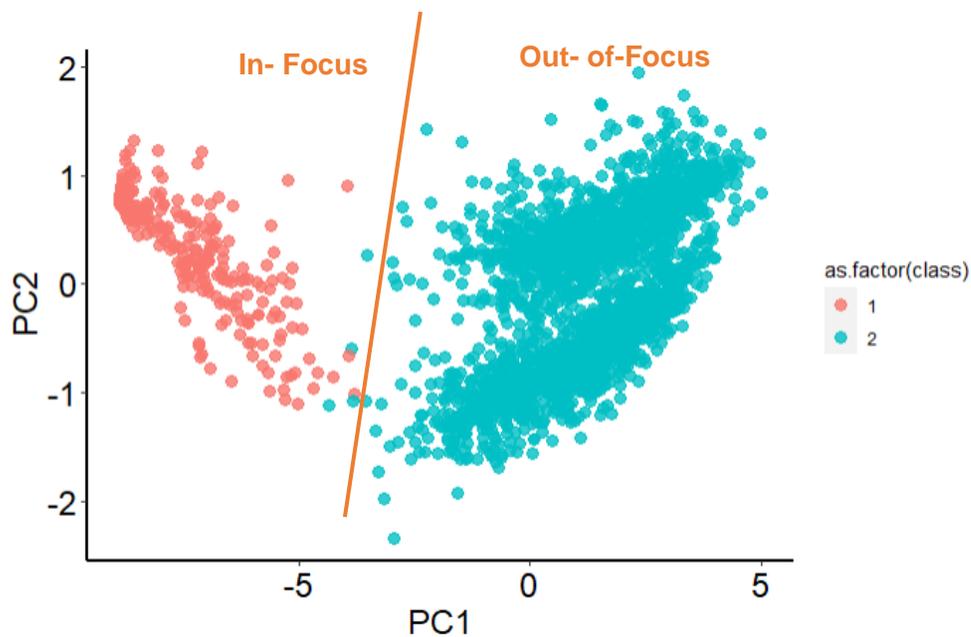


Figure 3.5 PCA Plot for In- and Out-of-Focus Clusters

3.4 Conclusion

Preliminary training with classification methods in both supervised and unsupervised learning demonstrated the advantage of FM over CNN: patterns were quickly and more explicitly recognized. The use of FM helped achieve 99.9% prediction accuracy of in- and out-of-focus images. However, the classification might not be the most suitable method for such a continuous distribution extracted by FM. As the number of classes increased, the performance of prediction deteriorates. The distance plot and matrix also showed this limitation because some inter-class Euclidean distances were indeed very small. Consequently, we decided to use non-linear regression in our training and applied a novel 3-stack method to distinguish both distances and directions of z-stacks.

Chapter 4 Feedforward Shallow Neural Network & Non-Linear Regression

4.1 Introduction

It was implied in the previous chapter that regression may fit our goal better, to generate predictions of continuous distances using focus measures. Chapter 4 introduced how we built a feedforward neural network with the nonlinear fitting of distributions of FM's. The hypothesis

$$Z = b + \omega_1 x_1^{n1} + \omega_2 x_2^{n2} + \omega_3 x_3^{n3} + \dots + \omega_{30} x_{30}^{n30}$$

Figure 4.1 Hypothesis Function of Non-Linear Regression Fitting

function (see Figure 4.1) demonstrates the relationship between predicted output (i.e. predicted distances) Z and predictors (i.e. values of focus measure) x_i . For example, x_1 is a vector of focus values of the first FM, so on, and so forth. n_i , w_i , and b are respectively power of predictors, weight (i.e. coefficient), and bias (i.e. intercept).

To measure the overall prediction error of the entire data set, Mean Squared Error (MSE) was used as the loss function, defined as the following, where m is the total number of input data points, z is the real recorded distance, and Z is the predicted distance.

$$MSE = \frac{1}{m} \sum_{i=1}^m (z_i - Z_i)^2$$

Figure 4.2 Equation of Mean Square Error^[7]

To minimize MSE, a method called gradient descent was applied in training optimization.^[7,8] The error curve for w or b is expected to have minima, where the gradient is zero. Starting at some randomly assigned parameter w or b , we moved with a step to “downhill” of the

“error well” based on the change of gradient. This process gets iterated until a local minimum was reached by recalculating the gradient to be zero. The step size is called learning rate and the error curve represents the loss function. More details about the gradient descent algorithm are shown in the table below.

<i>Gradient Descent Algorithm</i>
<p>1: Initialize random w and b</p> <p>2: Set initial epoch n = 1</p> <p>3: Set a learning rate α</p> <p>2: Repeat</p> <p>3: Compute loss function J, in this case, MSE</p> <p>4: Compute Partial derivative of J with respect to w and b</p> $\frac{\partial J}{\partial w} = \frac{1}{m} \sum_{i=1}^m ((wx_i + b) - z_i)(x_i^T); \frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^m ((wx_i + b) - z_i)$ <p>5: Update w and b</p> $b := b - \alpha \frac{\partial J}{\partial b}; w := w - \alpha \frac{\partial J}{\partial w}$ <p>6: n = n + 1</p> <p>7: Until $\ w_{n+1} - w_n\$ AND $\ b_{n+1} - b_n\$ are minimized</p> <p>8: Output the arrays of w and b</p>

Table 4-1 Gradient Descent Algorithm ^[7]

4.2 Methods

As we determined the model type and training algorithms, the problem of not being able to distinguish directions (e.g. above or below the focal plane) remains. As a result, we came up with a 3-stack method in image preprocessing besides normalization of focus values that was also applied in classification training in Chapter 3. The focus values of three stacks separated by equal distance were selected and numbered, where the most up one was labeled as “feature 1”, the middle one was “feature 2”, and the one taken at the lowest z-position was labeled as “feature 3” (see

Figure 4.3). Then the ratio of feature 1 to feature 2 and the one of feature 3 to feature 2 were calculated and compared (see Table 4-2). The comparison between these ratios would be enough to tell the direction of the middle stack due to the Gaussian distributions of most focus measures chosen in the training.

Comparison between two ratios	The direction of the middle stack
$\frac{\text{feature 1}}{\text{feature 2}} < \frac{\text{feature 3}}{\text{feature 2}}$	Above focal plane
$\frac{\text{feature 1}}{\text{feature 2}} = \frac{\text{feature 3}}{\text{feature 2}}$	Right at the focal plane
$\frac{\text{feature 1}}{\text{feature 2}} > \frac{\text{feature 3}}{\text{feature 2}}$	Below focal plane

Table 4-2 Comparison between focus value ratios predicts the direction

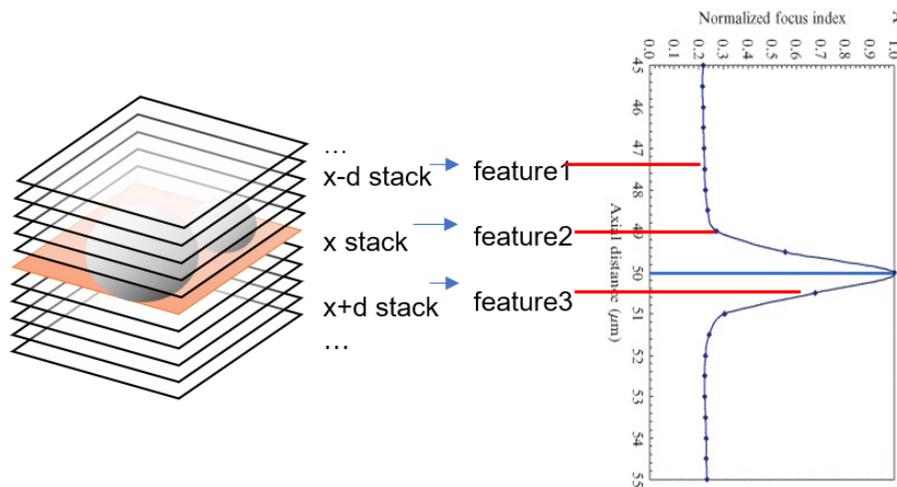


Figure 4.3 Extracting Ratios of Focus Values using 3-Stack Method

After the preprocessing of image data, we were able to distinguish both distances and directions of a stack by training a shallow neural network with two hidden layers and one output layer. Since each layer also has its weight and bias, the gradient descent is also applied to the optimization of parameters of layers. As the processed image data were input into the network, the

direction of the middle stack among the three stacks would be quickly recognized based on ratios and the distance would also be predicted through a nonlinear fitting. Note that there were about 680 sets of 35-stacks used in training and independent data, taken at a different time, used in testing. The training would stop when MSE stayed stable.

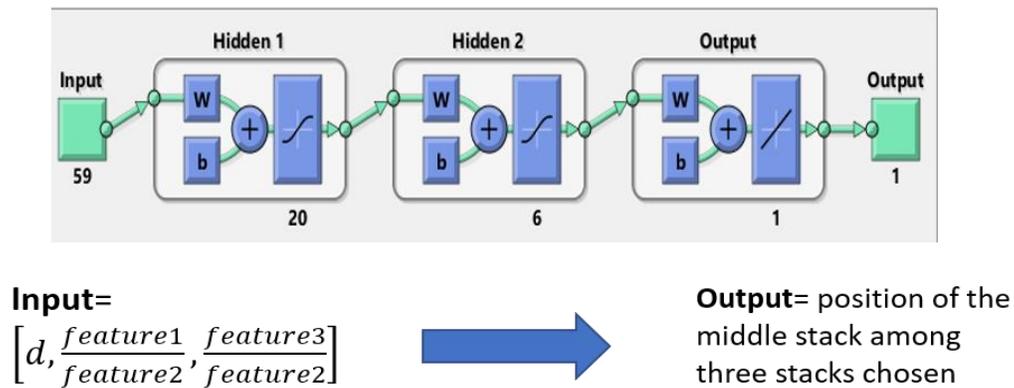


Figure 4.4 Feedforward Shallow Neural Network Structure

4.3 Results

As it was mentioned in the introduction of this chapter, MSE was used in evaluating the performance of both training and testing (a.k.a. validation). Compared with MSE, the Root Mean Square Error (RMSE) may be better to describe the average error of distance. Figure 4.5 showed one successful training with its training error and validation RMSE reduced to 0.34 micron. The agreement between training and validation demonstrates a robust fitting of our model. This model was then trained by the best combination of FM, which will be discussed with more details in Chapter 5.

We also defined an accuracy by counting the number of stacks whose individual error was within 1 micron. It was shown that 98.73% of predictions of stacks in training data were considered as accurate and 87% of independent predictions were thought to be so based on this criterion. We used the same network but with only OLDG to train the same data set again. It was shown that both training accuracy (around 94%) and independent testing accuracy (around 75%) were lower. Finally, to ensure that the prediction accuracy of stacks at each position (-17 um to +17 um) was consistent with the overall accuracy, we plotted the comparison between the predicted position and annotated position corresponding to each stack in our training data set (see Figure 4.6). The diagonal demonstrated that at each stack, predicted positions were mostly consistent with their real distances.

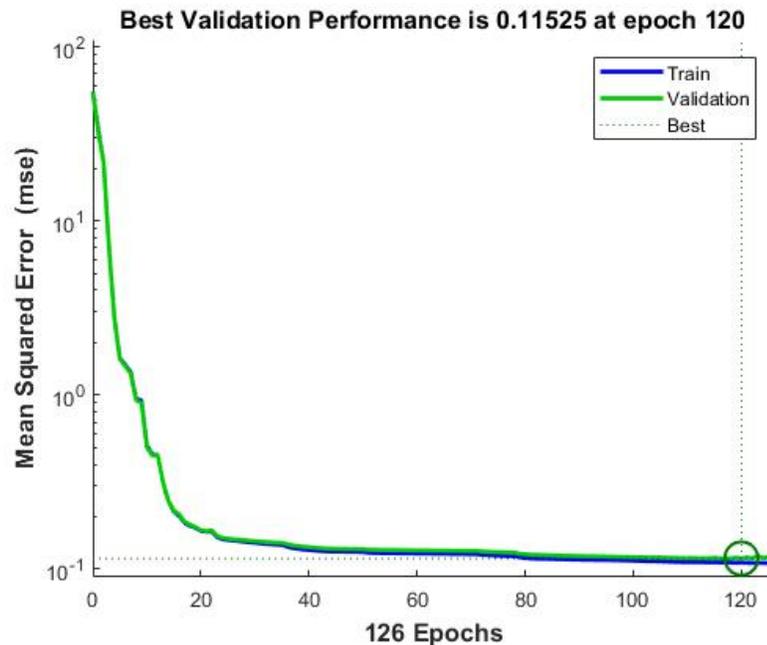


Figure 4.5 Training Curve of Best Combo of FM

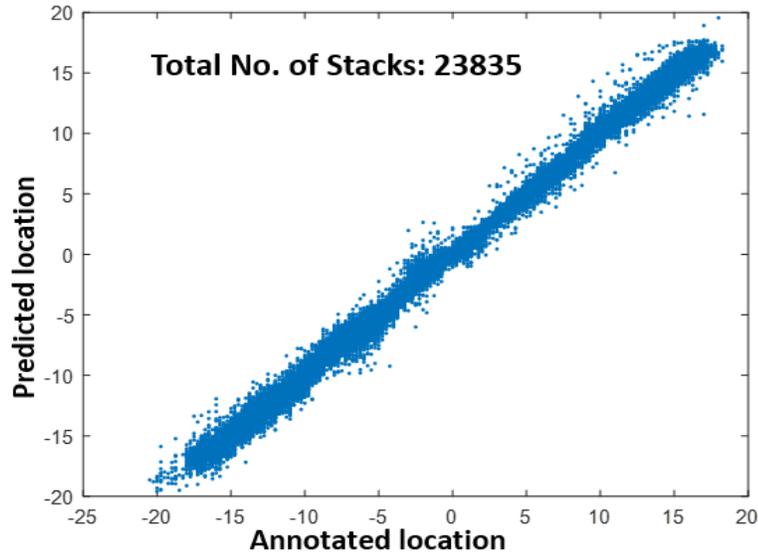


Figure 4.6 Comparison between Annotation and Prediction

4.4 Conclusion

In summary, the nonlinear regression implemented in feedforward neural network, based on gradient descent optimization, was a robust model for predictions of focus in terms of Gaussian distributed focus measures. It can output continuous distances within a relatively large range (e.g. -17 μm to 17 μm), with high prediction accuracies in both validation and independent testing. The use of combined focus measures was shown to have a better performance than the old, gradient-based feature. Most importantly, picking three equally spaced stacks from a set of 35-stack images and comparing their ratios of FM values allowed us to distinguish the direction of the middle stack, whether it was above or below the focal plane. The success of 3-stack method also implied that training might be done with much fewer stacks at one position, but the in-focus/middle stack must remain because others were normalized by the corresponding maximum. Such training can be done in the future to check if the prediction accuracy decreases.

Chapter 5 Feature Engineering

5.1 Introduction

This chapter demonstrated a variety of methods used in feature selection, including backward selection, feature importance, and Pearson correlations. A function in R, `VarImp` on package ‘`caret`’, regarding the sum of accuracy decrease as more variables are added, was used to evaluate the importance of each FM.^[9] Backward selection is applied manually to find the actual features in the combination that provides the highest prediction accuracy by using a for loop. Prediction accuracies given by cross validations are recorded as features were removed one by one according to the rank of feature importance. After the best-combined FM are determined, the potential collinearity problem was also examined through Pearson correlations.

5.2 Feature Importance in Classification

The importance of each FM was ranked in descending order and a sharp decrease of importance measure was found between the 16th and the 17th feature. As it was shown in Figure 5.1, the first sixteen highest-ranked features all have importance above 80%. Starting with the first sixteen features, multiple trials of training with ten-time cross validations were performed as features are removed one by one. All possibilities, including binary-class, 5-class, 6-class, and 7-class, were considered to generate curves of “average prediction accuracy vs. number of FM” (see Appendix E & F). It was demonstrated that the 13-FM combination tends to produce the highest accuracy, which is larger than the ones from 30-FM combination and OLDG

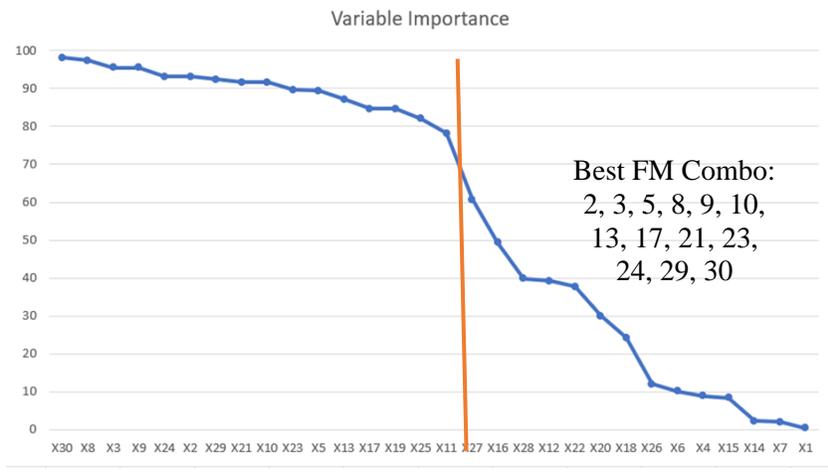


Figure 5.1 Rank of Variable Importance in KNN

5.3 Individual Feature Evaluation in Regression

Individual performance of each feature was evaluated regarding prediction accuracy generated by every single-FM training with regression, meaning that only one FM was used as the predictor. Figure 5.2 showed the percentage of accurate prediction (i.e. error within 1 micron) of all thirty FM's, some of whose performances were better than OLDG. These were selected and combined in further training, which produced better performances shown in Chapter 4.

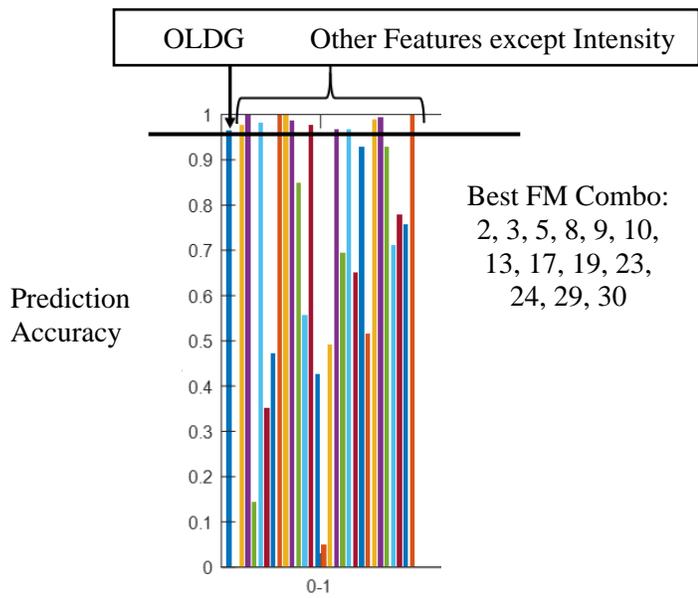


Figure 5.2 Individual Prediction Accuracies of Thirty FM's

Moreover, the highest-ranked FM's in classification and regression were generally consistent even though the ranking criteria were slightly different. It was ensured that these selected FM's were more "important" than others, generating both higher individual overall prediction accuracies.

5.4 Collinearity Check & Implications

Before I made conclusions, it was necessary to check the collinearity to get rid of variable redundancy. As a result, the Pearson correlation map was generated with features selected by previous steps. It was not surprising to see many high correlations among different variables (see Figure 5.3) according to the highly comparable "bell-shaped" distributions of our selected features. However, they should not be considered as redundant variables regarding the results from backward model selection – a certain combination of collinear variables still generate the highest prediction accuracy so far. A more comprehensive model selection, incorporating all possibilities of every different tuning parameter and FM can be done for further research. This can be accomplished in MATLAB using multiple nested for loops to go through every possible setup and to verify our feature evaluations discussed in this chapter before we implemented the best combination of FM's.

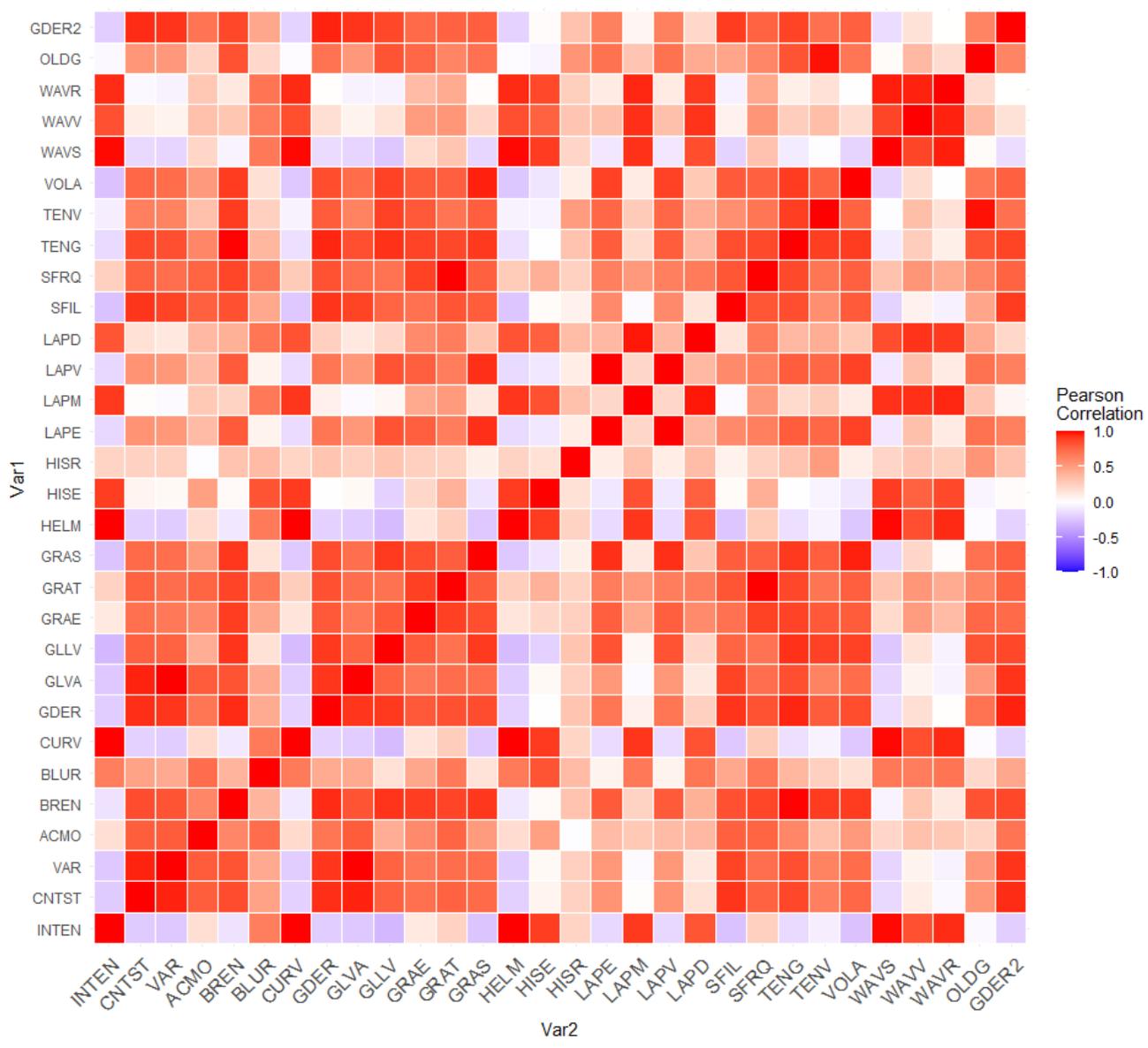


Figure 5.3 Pearson Correlation Map for Thirty FM

Chapter 6 Summary & Future Directions

This thesis demonstrated the exploration of improving software autofocus of phase-contrast images of yeast cells. CNN was first built with a regression layer to perform a task on predict continuous positions where images were taken. It was shown that CNN may not extract and prioritize features from 100-fold magnification images with a very small separation (0.05-micron step size) among stacks. Then thirty pre-recognized features -- focus measures -- were extracted from 40-fold images with 1-micron separation and used as predictors. Training with classification and regression was performed and validated to select the best-combined features. It was demonstrated that the selected combined features could provide higher prediction accuracies than the original software autofocusing did. It was also demonstrated that the 3-stack method was critical in predicting the direction of positions.

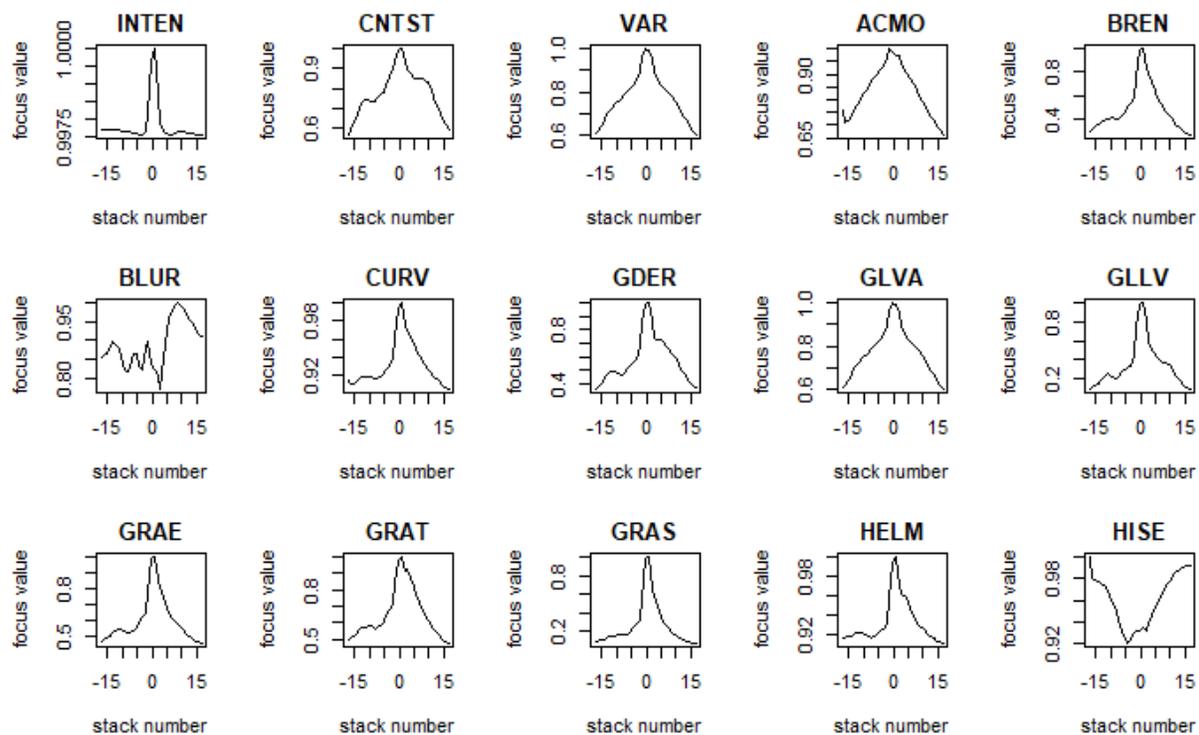
Such methods in machine learning and autofocusing can be applied in many other areas (e.g. autofocusing of animal cells), but there is yet much to be done about the refinement. First, CNN may be re-evaluated with simpler tasks as it was done for machine-learning classification and regression. Second, the success of 3-stack method in shallow neural network of regression indicates that fewer stacks (much less than 35-stack) can be sampled and used in training. As a result, such training can be done to compare with the one demonstrated in Chapter 4. Moreover, there are still many tuning parameters, such as the size of image downsampling, the number of stacks chosen for predicting one stack (e.g. 5-stack or 7-stack), the size of hidden layers of the neural network, etc. Further studies will be done to incorporate all these variables in more comprehensive training. It is suggested that this can be done with multiple “nested for loops” but such tasks require supercomputers or GPU to expedite the training process otherwise it may take

days or weeks as some preliminary training have already estimated. Finally, the new, optimized software will then be implemented into TimeLapse to improve autofocusing.

Appendix A Table of Details of Thirty Focus Measures ^[2]

Number	Abbr.	Full Name
1	INTEN	Intensity
2	CNTST	Contrast
3	VAR	Variance
4	ACMO	Absolute Central Moment
5	BREN	Brenner's Function
6	BLUR	Image Blurriness
7	CURV	Image Curvature
8	GDER	Gaussian Derivative
9	GLVA	Gray Level Variance
10	GLLV	Gray Level Local Variance
11	GRAE	Energy of Gradient
12	GRAT	Threshold Gradient
13	GRAS	Squared Gradient
14	HELM	Helmi's Mean Method
15	HISE	Histogram Entropy
16	HISR	Histogram Range
17	LAPE	Energy of Laplacian
18	LAPM	Modified Laplacian
19	LAPV	Variance of Laplacian
20	LAPD	Diagonal Laplacian
21	SFIL	Steerable Filters
22	SFRQ	Spatial Frequency
23	TENG	Tenegrad
24	TENV	Tenegrad Variance
25	VOLA	Vollath's Correlation
26	WAVS	Sum of Wavelet Coefficients
27	WAVV	Variance of Wavelet Coefficients
28	WAVR	Range of Wavelet Coefficients
29	OLDG	Old Method based on Gradient
30	GDER2	Gaussian Derivative 2

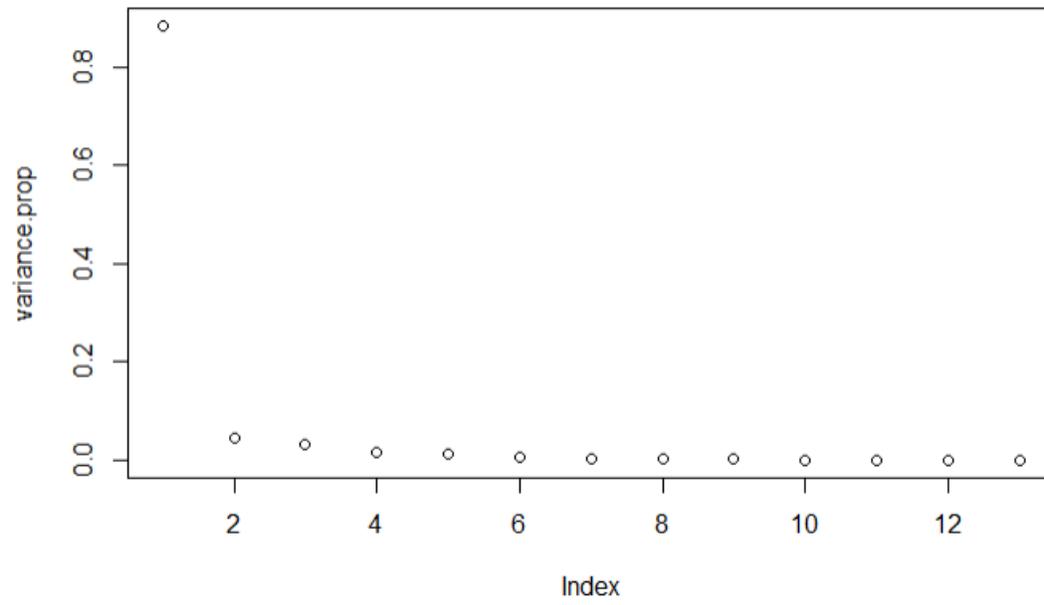
Appendix B Distributions of the Other Fifteen FM's



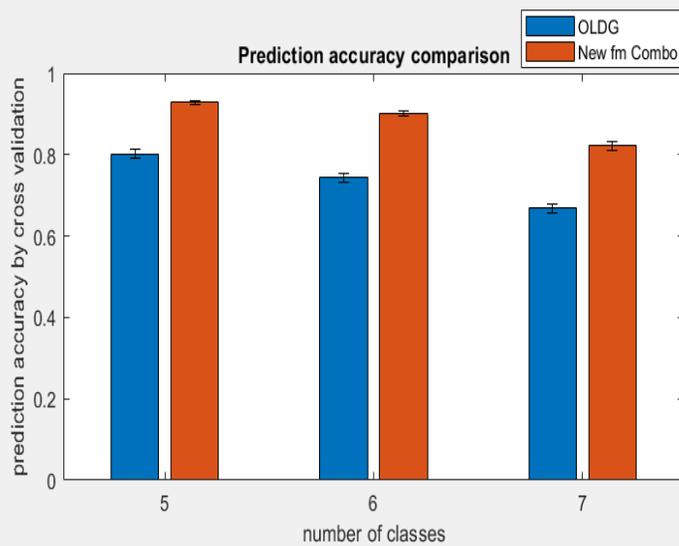
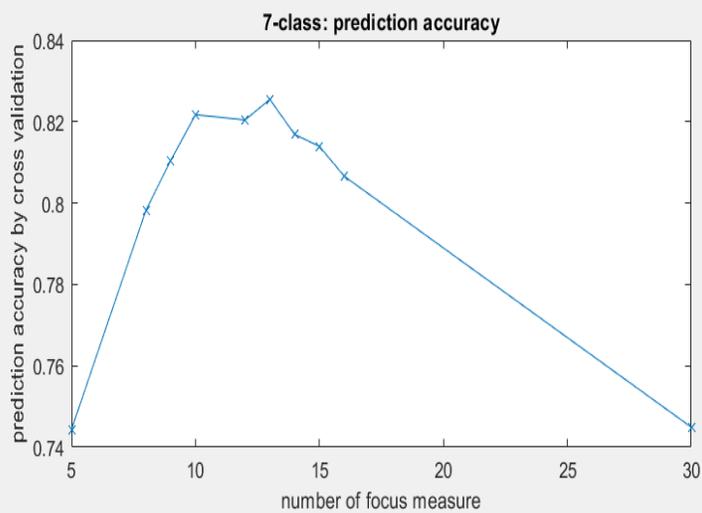
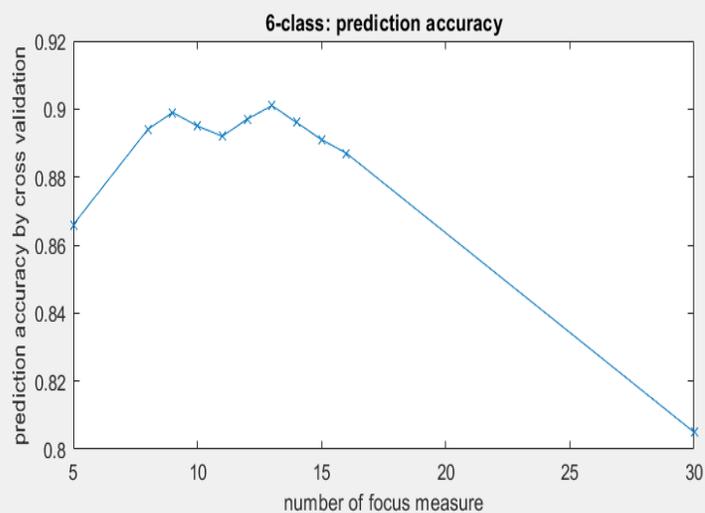
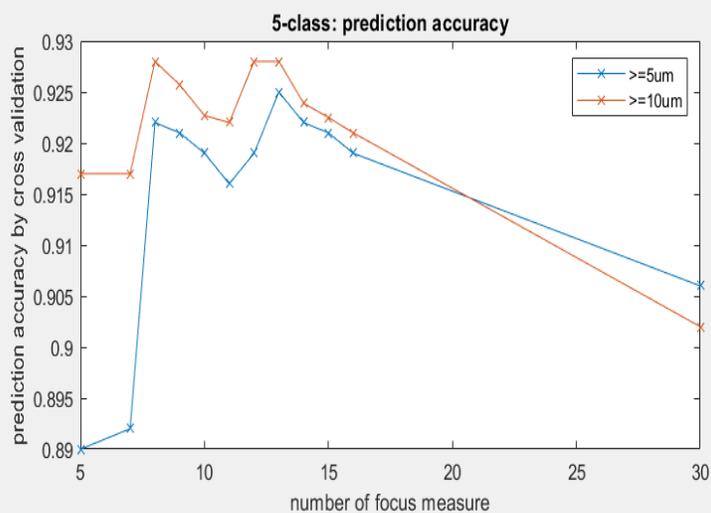
Appendix C Corresponding Classes for Multi- and Binary-Class Training with Classification

Distance (d)/micron	5-class (a)	5-class (b)	6-class	7-class	Binary
$0 \leq d \leq 1$	Class 1	Class 1	Class 1	Class 1	In-focus (1)
$1 < d \leq 2$	Class 2	Class 2	Class 2	Class 2	×
$2 < d \leq 3$	Class 3	Class 3	Class 3	Class 3	×
$3 < d \leq 4$	Class 4	Class 4	Class 4	Class 4	×
$4 < d \leq 5$	×	×	Class 5	Class 5	×
$4 < d \leq 10$	×	×	×	×	×
$5 < d \leq 10$	×	×	×	Class 6	×
$d > 4$	Class 5	×	×	×	×
$d > 10$	×	Class 5	Class 6	Class 7	Out-of-focus (2)

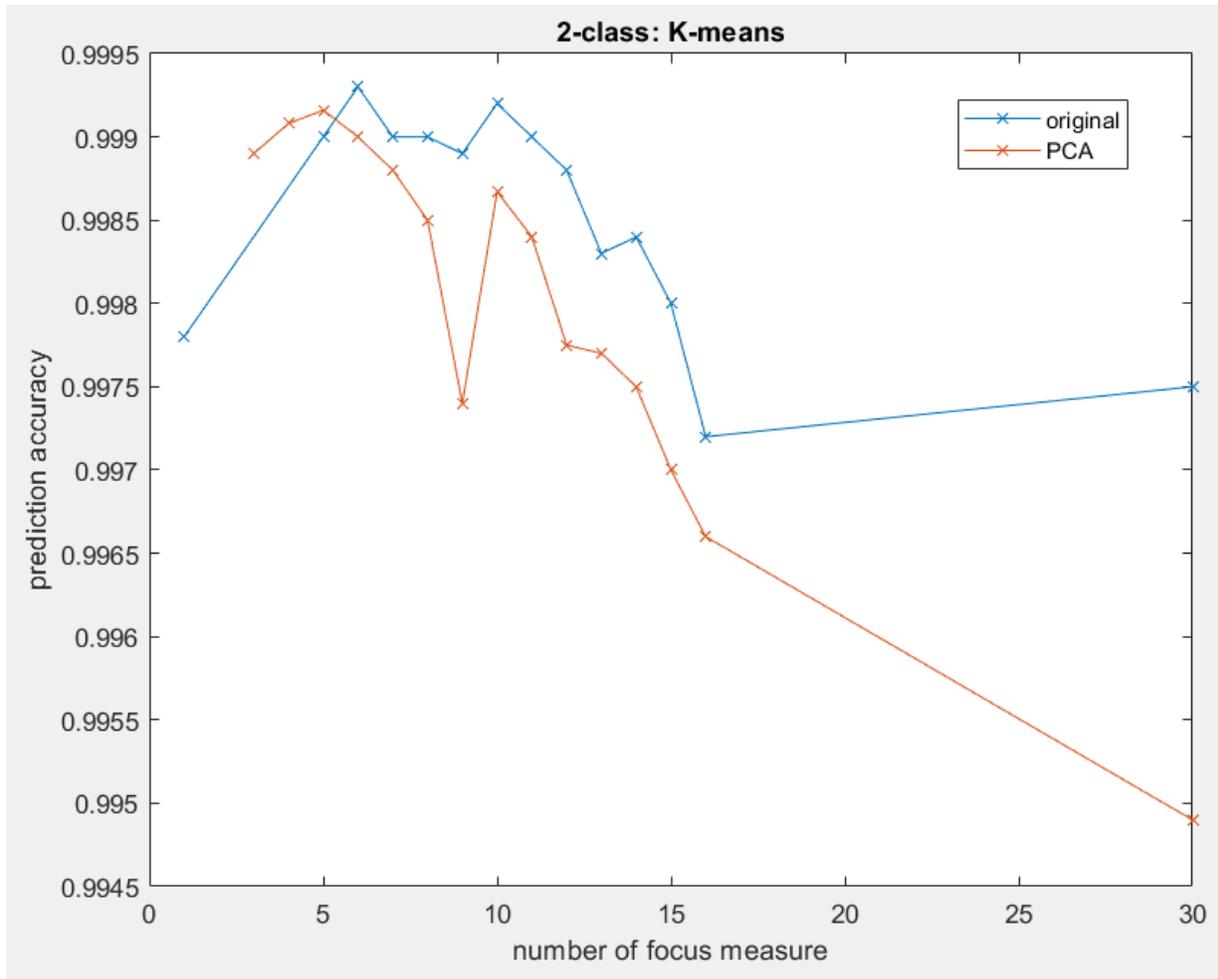
Appendix D Variance Plot of Principal Components



Appendix E Training Performance in Multi-class Classification with Different Combinations of FM



Appendix F Training Performance in Binary-class Classification with Different Combinations of FM



Bibliography

1. Zou, F., & Bai, L. (2019). Using time-lapse fluorescence microscopy to study gene regulation. *Methods*, 159-160, 138-145. <https://doi.org/10.1016/j.ymeth.2018.12.010>
2. Pertuz, S. (2017). Focus Measure. *MATLAB Central File Exchange*.
<https://www.mathworks.com/matlabcentral/fileexchange/27314-focus-measure>
3. Albawi, S., Abed Mohammed, T., & Alzawi, S. (2017). Understanding of a Convolutional Neural Network. *IEEE 2017*. doi:10.1109/ICEngTechnol.2017.8308186.
4. MathWorks. (n.d.). *Train Convolutional Neural Network for Regression*. Retrieved on February 25, 2021 from <https://www.mathworks.com/help/deeplearning/ug/train-a-convolutional-neural-network-for-regression.html>
5. Wei, L., & Roberts, E. (2018). Neural network control of focal position during time-lapse microscopy of cells. *Nature Scientific Report*, 8, 7313. <https://doi.org/10.1038/s41598-018-25458-w>
6. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. doi: 10.1007/978-1-4614-7138-7.
7. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.
<https://dl.acm.org/doi/book/10.5555/3086952>
8. MathWorks. (n.d.). *Feedforwardnet*. Retrieved on February 25, 2021 from
<https://www.mathworks.com/help/deeplearning/ref/feedforwardnet.html>
9. Kuhn, M. (2019). *The caret Package*. Retrieved on February 25 from
<https://topepo.github.io/caret/>

ACADEMIC VITA

YUJIE YAN

EDUCATION

Harvard Medical School

Major: Master of Biomedical Informatics

Anticipated: August 2021

Pennsylvania State University

Major/Option: Bachelor of Science (Honor), Physics/Medical

Minor: Biomedical Engineering, Mathematics

Academic Advisor: Prof. Richard Robinett

Expected Graduation: May 2021

HONORS/AWARDS

- Pennsylvania State University **Dean's List** **Dec 2017 – Present**
- Downsborough Department Head's Chair in Physics **Scholarship** **May - Aug 2019**
- Elsbach Honors **Scholarship** Physics **Aug 2019 – May 2020**
- J & E Teas – Science **Scholarship** **Aug 2020 – May 2021**
- **Inductee** of Penn State Sigma Pi Sigma Chapter **Mar 2020-present**

PUBLICATION

- Du, M., Zou, F., Yan, Y. & Bai, L. (2020). Chemically Induced Chromosomal Interaction (CICI): A New Tool to Study Chromosome Dynamics and Its Biological Roles. *bioRxiv*. 10.1101/2020.01.01.892448.

RESEARCH EXPERIENCE

Honor Thesis | Penn State University

Aug 2019 – Present

- Collaborate with Dr. Lu Bai and another PhD student on my honor project about developing and refining machine-learning models of microscopic autofocus in MATLAB
- Use optical microscope and TimeLapse to take z-stack, phase-contrast images of yeast cells
- Build a shallow neural network with regression layer in MATLAB to distinguish the positions of yeast cell images taken at different distances from the focal plane
- Extract features of z-stack images using focus-measure methods and evaluate different combinations of these methods
- Explore a variety of classification methods on distinguishing in-focus and out-of-focus images using both supervised and unsupervised learning in R

Research Assistant | Penn State University

June 2020 – Present

- Collaborate with Dr. Lu Bai and other PhD students and published a paper about Chemically Induced Chromosomal Interaction
- Use fluorescence microscope to take images of yeast cells

- Analyze data of fluorescence images of yeast cells using basic statistical methods

REU | Penn State Physics Department

May - Aug 2019

- Explore a deep-learning model, Convolutional Neural Network with regression, in MATLAB, in order to test if this model is competent to distinguish images taken at different axial positions
- Take images of yeast cells, analyze the image data, and calculate the co-localization fraction of chromosomes for Dr. Manyu Du's research

Research Assistant | Penn State University

Dec 2018 – Dec 2019

- Work in a condensed-matter lab with Dr. Eric Hudson and help with the deconstruction of Scanning Tunneling Microscope
- Explore the basic operation of 2D material imaging using STM

RESEARCH PRESENTATION

- Slide talk and poster presentation: Yan, Y., Zou, F., Du, M. & Bai, L. (2019) Autofocus of Phase-Contrast Images of Yeast Cells using Convolutional Neural Network. Penn State University REU 2019 Symposium.
- Poster presentation: Yan, Y., Zou, F. & Bai, L. (2020) Autofocus of Phase-Contrast Images of Yeast Cells with Focus-Measure and Machine Learning. Conference for Undergraduate Women in Physics at Temple University. American Physical Society.

TEACHING EXPERIENCE

Undergraduate TA | Theoretical Mechanics

Aug 2020 - Present

- Collaborate with Dr. Eric Hudson and other undergraduate instructors on developing class materials of Theoretical Mechanics
- Guide collaboration of in-class activities to help them understand concepts better
- Help with office hours to solve homework problem and concerns in both academics and life
- Communicate with other instructors about reflections and improvements of classes on weekly meetings

Learning Assistant | Intro of thermodynamics, fluid & quantum mechanics

Sep - Oct 2018

- Work with Dr. Douglas Cowen to help students work through in-class problems

EMPLOYMENT

International Orientation Leader | Global Programs, Penn State University

Mar -

Aug 2019

- Assist as an academic, social, and personal resource
- Oversee 30 new international students as teams for two weeks to participate school-organized events/gathering

- Guide collaborative activities to help students get used to American university environments and facilitate communication among different international cultures

EXTRACURRICULAR

Student Space Program Laboratory | Penn State University

Sep 2018

- Participate in Student Training Program and cooperate to design a rocket for the detection of CO₂ concentration and UV radiation

Science Demo Performer | Art Festival, State College

Jul 2019

- Perform a science demo of liquid nitrogen to make frozen flowers, deflating balloons, and ice creams.
- Take care of 5 to 13-year-old kids to ensure their safety in an environment of science experiment

Member in Society of Physics Students | Penn State University

Aug 2017- Present

- Attend weekly meetings and social networking with other physics majors

SKILLS

- **Programming:** MATLAB (advanced), R(advanced), Python (NumPy, TensorFlow)
- **Technical Tools:** LaTeX; Microsoft Office; Photoshop; Supercomputer ROAR (at Penn State)
- **Language:** Competent in written and spoken English & Chinese
- **Lab:** General physics, chemistry, molecular biology, and material science lab skills
- **Other:** Watercolor; Traditional Chinese Painting and calligraphy; playing guitar