Automated Image-Based Quality Control of Molecularly Imprinted Polymer Films

by

© Andrew R. Way

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science.

Graduate Program in Scientific Computing

Memorial University

October 2019

St. John's, Newfoundland and Labrador, Canada

Abstract

We present results of applying a feature extraction process to images of coatings of molecularly imprinted polymers (MIPs) coatings on glass substrates for defect detection. Geometric features such as MIP side lengths, aspect ratio, internal angles, edge regularity, and edge strength are obtained by using Hough transforms, and Canny edge detection. A Self Organizing Map (SOM) is used for classification of texture of MIP surfaces. The SOM is trained on a data set comprised of images of manufactured MIPs. The raw images are first processed using Hough transforms and Canny edge detection to extract just the MIP-coated portion of the surface, allowing for surface area estimation and reduction of training set size. The training data set is comprised of 20-dimensional feature vectors, each of which is calculated from a single section of a gray scale image of a MIP. Haralick textures are among the quantifiers used as feature vector components. The training data is then processed using principal component analysis to reduce the number of dimensions of the data set. After training, the SOM is capable of classifying texture, including defects.

Dedicated to my parents for their support of my education

Lay summary

In a mass production environment, determining the quality of something usually involves finding defects. Take, for example, finding defects in bottles of beer. A defect could be a crack in the glass, an unusually wide neck, or an abnormally cloudy beer. Almost anyone walking the street could be asked to perform the task of finding such defects with minimal training. The problem is that people get tired, bored, or unmotivated pretty quickly. Sooner rather than later, the quality control worker begins to miss an increasing number of defective bottles. Human eyes, though attached to highly intelligent brains, tire quickly.

As an alternative, machines with the aid of computers can be used to find those same defects. Computers are really good at just one thing: adding numbers. Since they know how to add, they can also subtract, multiply, and divide. They're also good at comparing numbers. Herein lies their strength; all operations performed by humans that are, at their root, basic arithmetic can be performed at inhuman rates by a computer. The video you stream on your computer, or the word processor you use at work are all provided to you by a computer performing basic arithmetic at unimaginable speeds. In addition to that, the memories of computers are virtually infallible; they do not forget what they know unless they are physically damaged or irradiated, or unless they are instructed by the user to forget it. They are the most reliable means of performing routine tasks day in and day out provided that we, the programmers, give them the right instructions. Providing this set of instructions such that the computer obtains sufficient independence and intelligence is the difficult task. In other words, machine eyes never tire but they can't think for themselves.

This is the challenge we set out to solve for at least one scenario, one involving films of molecularly imprinted polymers (MIPs). To the average person, a MIP is just a "smear of paste" spread on a piece of glass. These smears of paste must be sold as a commercial product and they must look good. In this thesis, we present our research on giving the machine enough intelligence to determine the degree of excellence of the MIP.

Acknowledgements

I would like to thank Memorial University of Newfoundland, ACOA, and NSERC for funding this research. I would like to sincerely thank Dr. Marek Bromberek for quickly supplying high quality imaging equipment, without which these results would not have been attainable. I would also like to thank Dr. Ali Modir for providing the MIPs. Furthermore, I would like to thank Dr. Mohamed Shehata of Memorial University of Newfoundland for providing initial source code for performing Hough transforms. Finally, I would like to thank my supervisors Dr. Carlos Bazan and Dr. Ivan Saika-Voivod for guiding this work.

Table of contents

List of tables

List of figures

Chapter 1

Introduction

1.1 Overview

Industrialization and urbanization have improved economies at the expense of clean natural water resources. In turn, they created the need for timely and efficient responses for minimizing the negative effects of resulting accidents, or natural disasters that may affect fresh water supply. As a consequence, there is an increased interest in affordable, fast and easy-to-use methods for routine *in situ* analysis and monitoring of markers of water contamination. Sensing systems that employ molecularly imprinted polymers (MIPs) are an attractive alternative for analysis of contaminants in aqueous media. In order to provide sufficiently high quality MIPs for commercialization, effective and efficient quality control processes must be put in place. In this thesis, we discuss such a system which consists of a computational pipeline of several methodologies based on image analysis.

1.2 Molecularly Imprinted Polymers

MIPs are synthetic materials that can be tailored to recognize specific compounds in complex samples by using molecular recognition similar to the "lock-and-key" mechanisms in biological receptors such as antibodies or enzymes [1][2][3]. Due to the MIPs molecular recognition abilities, MIPs are employed in applications involving the determination of pollutants in water [4]. The most successful application of MIPs involve solid phase extraction, a process known as molecular imprinted solid phase extraction (MISPE). Solid phase extraction is a process in which compounds that are suspended in a liquid mixture separate based on their physical and chemical properties. MIPs are used in tandem with this process. Xu [5] used MISPE with liquid chromatography to detect trace levels of a compound known as estrone in water, and achieved a limit of detection of 5.7 ng/L. Other applications of determination of pollutants include detection of water-soluble acid dyes [6], herbicides [7], and cyanide [8]. MIP films provide an effective and portable detection solution for these pollutants, which assists in mitigating their consequences on public health.

1.3 MIP Production

One of the most notable characteristics of MIPs is the simplicity of the fabrication process. This makes them an attractive component of water quality sensors that can be mass-manufactured by automated means. For their fabrication, MIPs are synthesized through polymerization of a functional monomer and a cross-linking agent around a template molecule (the compound of interest) [4] as shown in Fig. 1.1. Initial interactions take place in the pre-polymer mixture among the different components of the polymerization solution, in particular, the interaction between the template molecule and the functional monomer. When the template (or pseudo-template) is removed, we are left with a polymer showing complementary cavities in shape, size and functional groups to the target molecule. This is visible in the lower half of Fig. 1.1. The properties of the polymer can be tailored to suit specific compounds representing the water contaminant of concern.

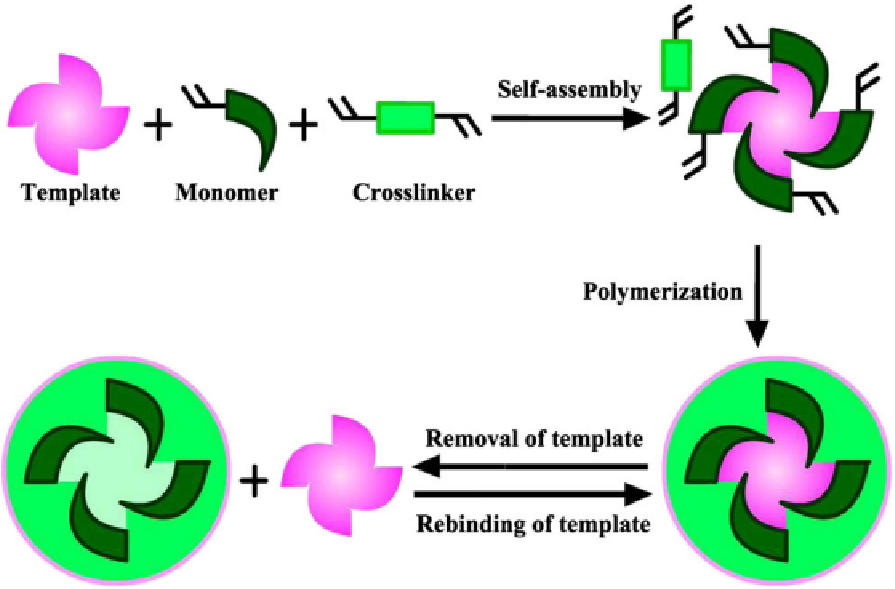


Figure 1.1: A flow chart illustrating the life cycle of a MIP film from production to usage. Reproduced with permission from Ref. [4].

1.4 Quality Control Demands

MIPs have been developed in a range of formats to suit various applications, e.g., monoliths, beads, micro-spheres, and surface imprinted films [4]. Using MIPs in film format has many advantages for use in sensors for direct measurement of the compound bound to the film. There are current efforts to provide portable, selective and sensitive solutions for the characterization of water contaminants of concern by using MIPs in water monitoring products [9]. Therefore, there is a need for devising quality control methodologies for the manufacturing of new synthetic materials that form part of the sensing products. The proposed quality control methodologies involve the development of image processing and analysis techniques.

1.5 Computational Approach to Quality Control

The computational approach employed in this quality control system for MIP films is based on combining several methodologies into a single pipeline, shown in Fig. 1.2, which takes a color image containing a MIP film and outputs texture and geometry analyses.

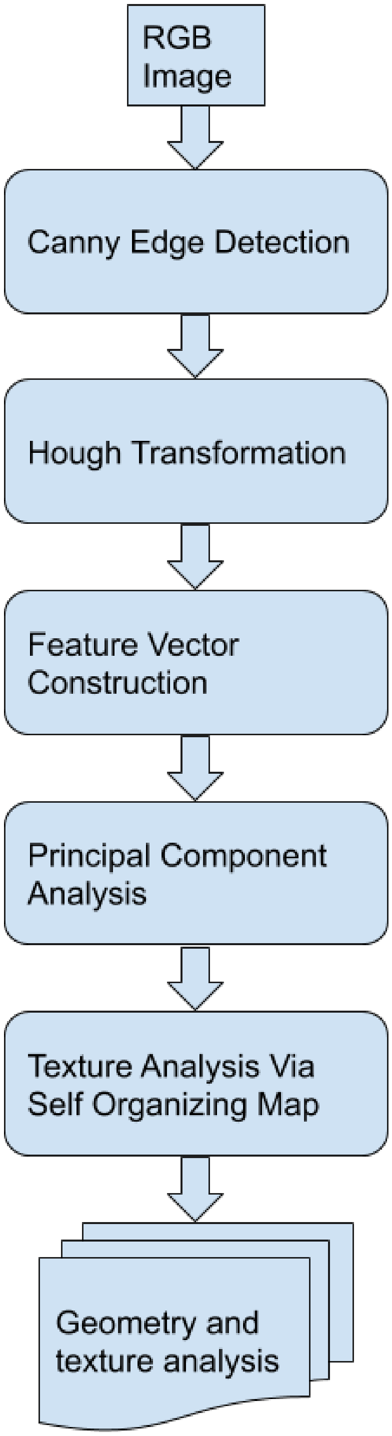


Figure 1.2: The computational pipeline used by the quality control system for MIP films.

The first step of the pipeline involves the Canny edge detector [10] which finds edges within an image using a multi-step method later described. The result is a black and white image where the white pixels correspond to edges in the original image. The next step in the pipeline is to apply Hough transforms, which finds straight lines, to the image obtained from the Canny edge detector.

Figure 1.3: Raw image of the top of a cargo truck. Reproduced with permission from Ref. [11].

As an illustration, we consider the example provided by Xie and Zhou [11] in which Canny edges and Hough transforms were used to aid in the automated detection of a truck cargo compartment shown in Fig. 1.3. The Canny edge image is shown in Fig. 1.4.

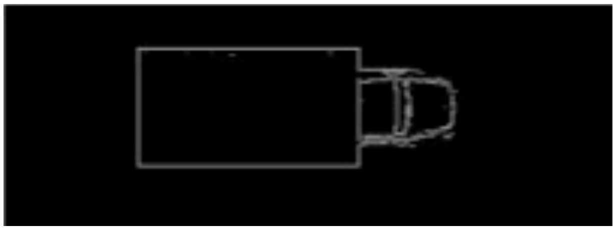


Figure 1.4: Canny edge image of the top of a cargo truck. Reproduced with permission from Ref. [11].

After the Canny edge image was obtained, it was processed using the Hough transform algorithm. Hough transforms find straight lines by generating a set of lines, with varying orientation and distance from the origin, and counting the number of edge pixels each line intersects. Thus, the line that intersects and is co-linear with a long straight edge in the Canny edge image will have a high pixel intersection count, while other lines which intersect the edge but are not co-linear will have lower pixel intersection counts. The problem Xie and Zhou solved was finding which four lines corresponded to the rectangle of the cargo compartment shown in Fig. 1.4. The rectangular compartment was found by finding a set of four lines which formed a rectangle.

These four lines appeared as peaks in the so-called "accumulator space" which displays the edge pixel intersection count for each possible line, defined by parameters ρ and θ. The accumulator space is visible in Fig. 1.5. ρ gives the perpendicular distance of the line from the origin of the image and θ defines the orientation angle of the line. For the compartment problem, this meant finding two pairs of lines that were parallel or, equivalently, having the same θ value. Additionally, the pairs themselves should be orthogonal to one another, which means having θ differing by 90°.

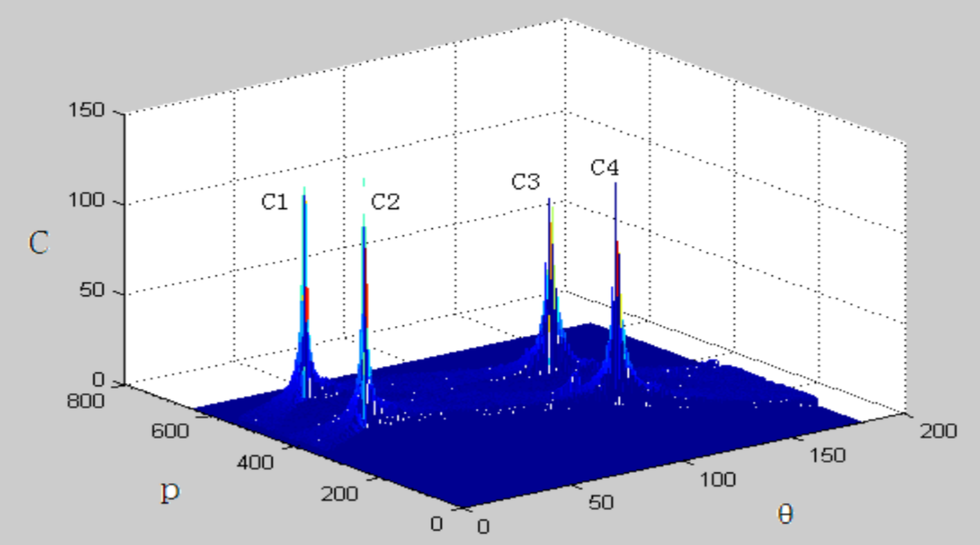


Figure 1.5: The accumulator space obtained from performing a Hough transform on a Canny image of a truck. The four peaks correspond to the four sides of the trucks rectangle compartment. Reproduced with permission from Ref. [11].

This problem of finding a rectangle from a set of lines given some geometric constraints is similar to the problem of detecting a MIP film within an image. MIP films have the constraint of being rectangular. Thus, by using a similar approach as Xie and Zhou, the rectangular outline of the MIP film may be found by finding which four lines create a four-sided polygon with a desired surface area, aspect ratio, and internal angles. Once the four lines are found, one may find the dimensions of the MIP film as well as remove parts of the image outside of the area enclosed to reduce the overall size of the image.

Once the geometry of the MIP film has been assessed, texture analysis is performed. Prior to performing texture analysis, each image needs to be translated into a feature vector, a collection of numbers calculated from features within the image. Zayed [12] used the well-known Haralick texture features in a statistical approach for discriminating lung abnormalities. In their study, they found that certain Haralick texture features were significantly different in images containing tumors versus edema (or swelling). The images used for the data set were of cross-sectional computerized tomography scans of healthy lungs, lungs with tumors, and lungs with edema. Haralick features were chosen to be used for this thesis with the intent of finding abnormalities in MIP film texture, as they have stood the test of time as effective general purpose texture descriptors [13][14][15][16][17].

Once all desired features are calculated for sections of each image, the data set for texture analysis is finally formed. The next step in this computational approach is to reduce the dimensionality of the data set by performing principal component analysis (PCA). The aim of PCA is to find a set of uncorrelated feature vectors that point along the directions of the greatest variation in the original data set. The idea is that the original set of feature vectors describing the data set do not necessarily point in the direction of maximal variance because different features, generally speaking, are correlated. Howley [18] used PCA in a machine learning application and explored the effect it had on improving prediction accuracy and preventing the common problem of over-fitting due to too many dimensions. Howley found that in particular the support vector machine (SVM) classifier, which finds hyper planes that separate classes of data, and the k-nearest neighbors (k-NN) classifier, which assigns a class label to a data point based on the class labels of surrounding data points, achieved better results more efficiently when PCA was used. The error for these classifier methods decreased to a minimum for approximately 8 principal components, and subsequently increased with an increasing number of principal components used as shown in Fig. 1.6.

In general, the number of principal components chosen for the final data set is typically done using rules of thumb. According to Rea [19], several standard methods exist, one of which is the scree plot test which involves taking all components up to the point at which there is an "elbow" in the plot of the variance as a function of number of principal components used [20]. Once the principal components have been identified, they are examined by a subject specialist to determine if the principal components have meaning. In this research, the scree plot method is used.

Finally, texture analysis is performed by taking the reduced set of feature vectors and using them to train a sheet-like artificial neural network known as a self-organizing map (SOM). A SOM contains cells that become tuned to specific input signal patterns

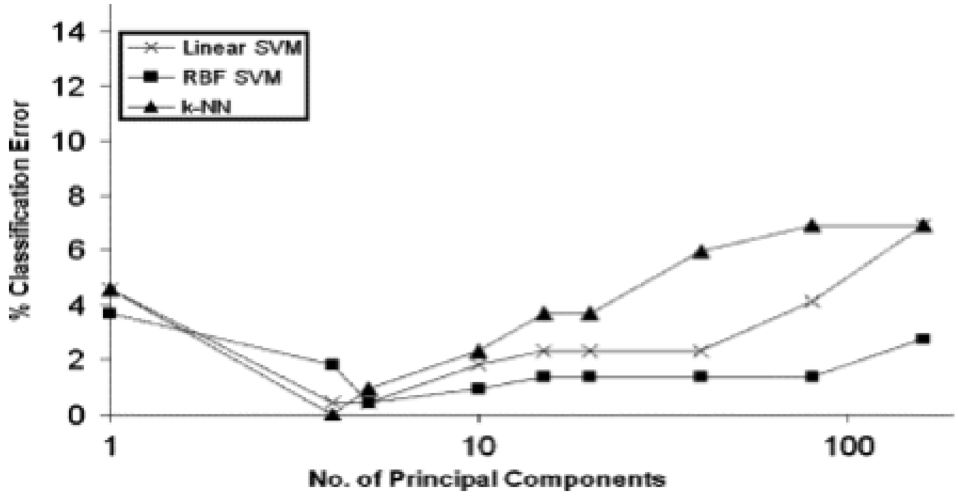


Figure 1.6: A plot of classification error, using a given machine learning algorithm, versus number of principal components. Reproduced with permission from Ref. [18].

through an unsupervised learning process. Once trained, the cells within a SOM tend to become spatially segregated in such a way that cells that respond similarly to a specific input signal will reside within a domain. The spatial segregation of cells based on response to input signal is similar to how topology of the brain is related to function. Specific sections of the brain, when stimulated, disturb the function that is associated with that section. The disturbance may induce or inhibit some function, such as the movement of the hand. In other words, responses to input signals are spatially organized within the brain's cerebral cortex. A SOM behaves similarly by having input signals induce a greater response with specific cells, or domains of cells, than others.

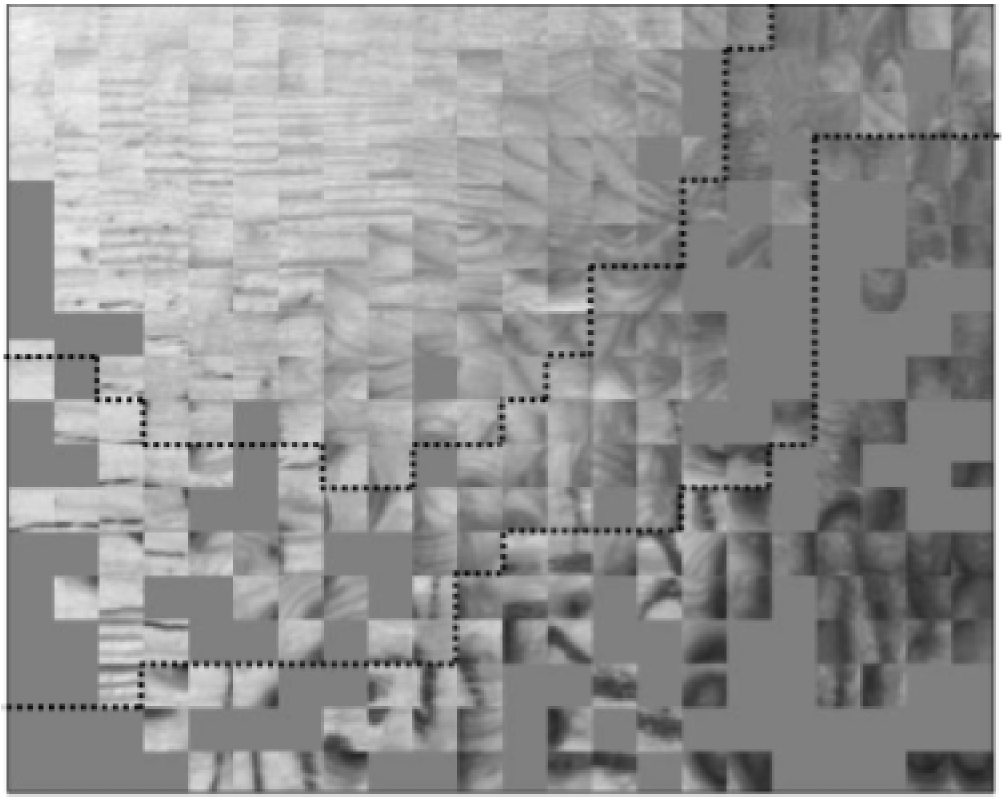


Figure 1.7: An illustration of a trained SOM, where each square's displayed image indicates the type of wood image cell that is mapped to the square's underlying neuron. Reproduced with permission from Ref. [21].

Niskanen [21] showed that a SOM could be trained on images of lumber to find and flag wood knots. The result of training the SOM is shown in Fig. 1.7. Each square cell corresponds to a neuron in the SOM, and the image of the wood shown is an example of an image cell that is mapped to that square. Evidently, wood knots tend to congregate near the bottom right corner of the SOM. Boundaries were created in the SOM for future classification, where if a new image cell was mapped within the lower-right boundary it could be classified as a wood knot. Thus, a SOM may be used for classification by assigning each domain a particular label. In the scenario of using MIP films, the usage will be similar; image cells of similar texture, such as scratches, should congregate in one section of the SOM. Subsequent to training, future image cells falling into that section of the SOM will be classified as scratches.

In addition to the literature reviewed above, a plethora of examples of applications of machine learning using image data are reported in the literature. In particular, machine learning finds extensive usage in quality control applications in the agricultural and fishing sectors, such as detecting the freezer burn rates in salmon via the TreeBagger classifier algorithm [22], detecting defects in raw potatoes using a convolutional neural network [23], measuring stress severity in soybeans using several different algorithms [24], and differentiating walnut shells from walnut pulp [25]. In the medical fields of research, back-propagation neural networks have been used for recognizing major sub-cellular structures in fluorescent microscope images of HeLa cells [26]. Additionally, quality identification of human induced pluripotent stem cell colony images have been achieved using support vector machines [27]. Finger print quality assessment has also been achieved using a SOM in-tandem with a random forest [28]. Finally, applications of machine learning on image sets exist in the mining and materials engineering fields as well. Applications include determining limestone ore grades using PCA and a neural network [29], recognizing patterns on images of metallurgical micro-structures using a random forest [30], and finally, the detection of residual traces of coating produced in metal welding via support vector machines [31].

1.6 Overview of Thesis

The remainder of this thesis is as follows. Chapter 2 is dedicated to describing the computational methodologies and features used to create the geometric and texture analyses of the computational pipeline. It describes how the images are obtained, how the MIP film region is detected and measured using Canny edge detection and Hough transformations, how the quality of the MIP film is measured, how the texture features are calculated prior to using PCA, and how the SOM is used for texture analysis. Chapter 3 is dedicated to results on applying the computational pipeline to generate the geometric and texture analyses. Additionally, results on the effects of changing certain parameters, like the number of neurons in the SOM, are presented. Chapter 4 includes discussion on the performance of the quality control pipeline. Finally, Chapter 5 discusses and summarizes the results and outlines potential for future work.

Chapter 2

Methods

2.1 Image Acquisition

Images of the MIP films were taken with a Canon EOS5D II camera shown in Fig. 2.1a, a Canon 580 EX 2 Speedlight light source shown in Fig. 2.1b, and a Cactus v5 radio trigger for remotely triggering the image capture process. The camera used a Canon EF 100mm f/2.8 macro lens. The MIP films were placed on a large piece of glass. An example of a MIP film on this glass is shown in Fig. 2.2. Immediately below the MIP film is a ruler which was used to provide a scale for the quality control system. Behind the supporting piece of glass was the light source shown in Fig. 2.1b. Once the button on the remote was clicked, the light source illuminated the underside of the glass slide and then the picture was acquired. All metadata for the images, such as camera settings, are found in App. A.

(a) Canon EOS5D II [Source: Charles Lanteigne, https://commons.wikimedia.org/wiki/File:Canon_EOS_5D_Mark_II_with_50mm_1.4_edit1.jpg]

(b) Canon 580 EX 2 Speedlight. [Source: KaiAdin, https://commons.wikimedia.org/wiki/File:Canon_Speedlite_580EX_II_Front.jpg

Figure 2.1: Image acquisition equipment.

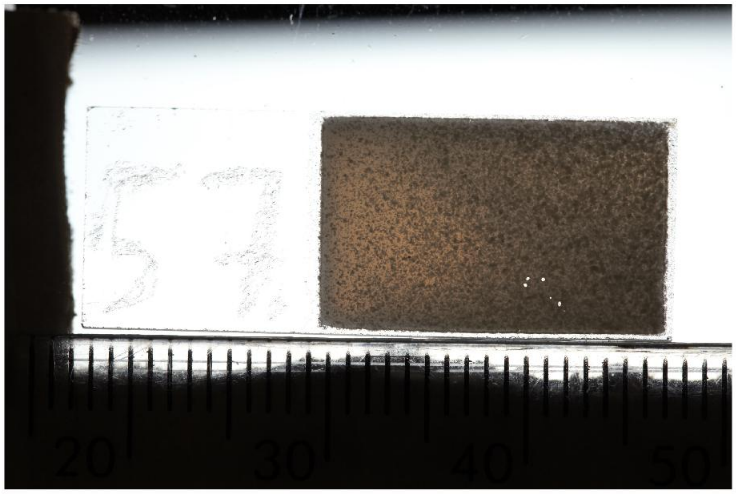


Figure 2.2: An example of a MIP film imaged using the described image capture setup.

2.2 Canny Edge Detection

Canny edge detection [10] provides structural information about an image while suppressing noise. The multi-step process is as follows:

1. Noise reduction using a Gaussian filter
2. Calculation of intensity gradients within the image
3. Non-maximum suppression
4. Double threshold
5. Hysteresis-based edge tracking

Below, we explain each of these steps.

2.2.1 Gaussian Filtering

Since edge detection results are easily affected by noise, Gaussian filtering is applied to the images to perform noise reduction. Gaussian filtering works by first constructing a Gaussian kernel of size $(2k+1)\times(2k+1)$ given by

$$G_{ij} = \frac{1}{2\pi\sigma^2} \exp - \frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}; 1 \leq i, j \leq (2k+1) \tag{2.1}$$

and performing a type of averaging over pixel intensities with weights determined by the Gaussian kernel. For this thesis, $\sigma = \sqrt{2}$ and $k = 6$, and the Gaussian kernel is normalized such that its elements sum to one. Before filtering, the red-green-blue (RGB) image is first converted to a gray scale image by calculating a weighted sum of the RGB components using the formula

$$GL_{ij} = 0.299R_{ij} + 0.587Gr_{ij} + 0.114B_{ij} \tag{2.2}$$

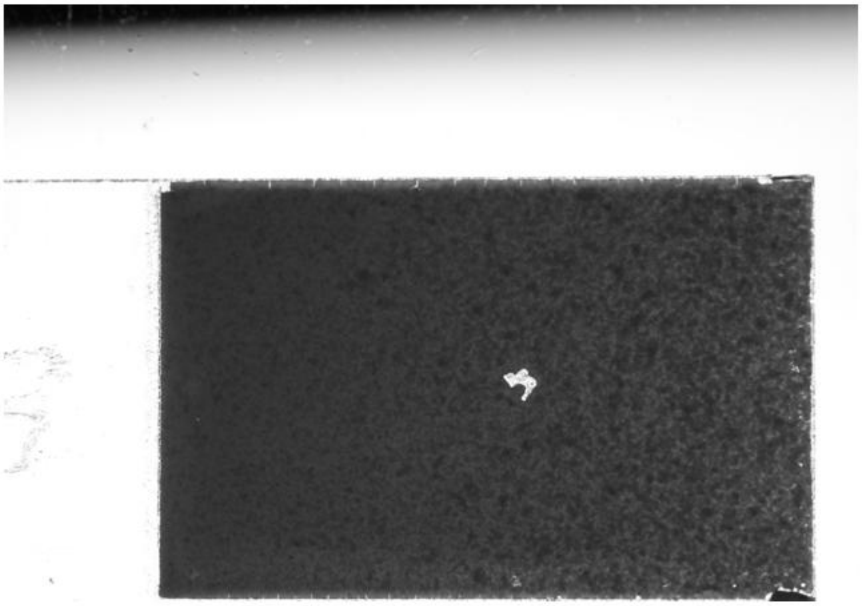
where GL_{ij} is the gray level, and R_{ij}, Gr_{ij}, and B_{ij} are the components for each of the RGB channels for the pixel at row i and column j. This calculation eliminates the hue and saturation information of the original image while retaining the luminance. Gaussian filtering is then performed by moving the kernel across the gray scale version of the image and performing a convolution operation on each underlying $(2k+1)\times(2k+1)$ subsection of the image $\mathbf{I}$, with the result being expressed as $\mathbf{I}'$.

$$\mathbf{I}' = \mathbf{G} * \mathbf{I} \tag{2.3}$$

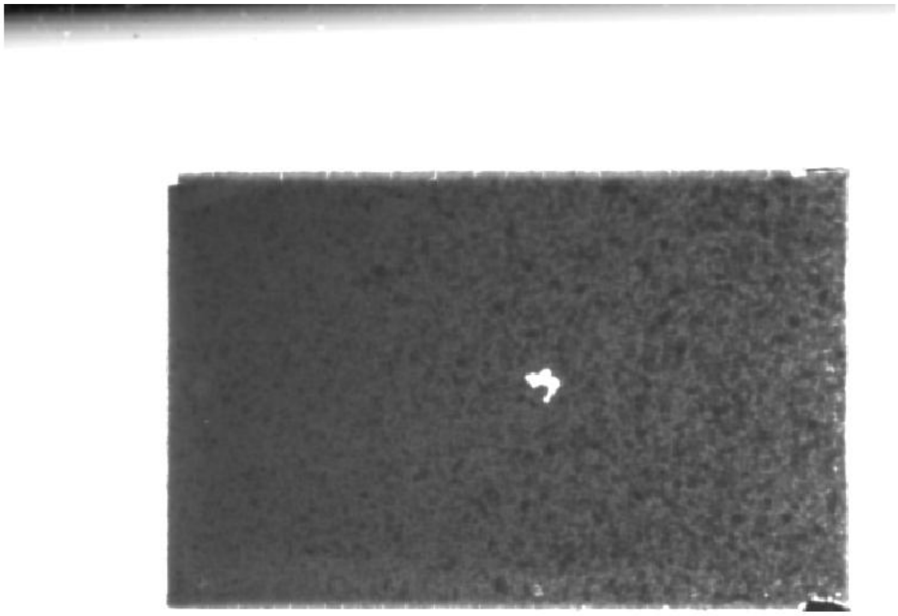
For example, a 5×5 Gaussian kernel is chosen to smooth the image shown in Fig. 2.3a. The kernel, which is just a matrix, is given by,

$$\mathbf{G} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \tag{2.4}$$

The convolution operation is then performed on each 5×5 subsection of the image which results in the image shown in Fig. 2.3b. The most noticeable change between the two in this particular example is that the rough texture of the underlying glass slide dissipated.



(a) An un-processed MIP film.



(b)

Figure 2.3: An example of Gaussian kernel-based filtering on an image of a MIP film.

The differences between the original image and the filtered image may be observed in Fig. 2.4, in which the same top right corner of the film from both Fig. 2.3a and Fig. 2.3b are expanded and compared side-by-side. The most noticeable difference is the removal of the edge pixels from the underlying glass slide, and the decreased spatial variation in pixel intensity in the filtered image.

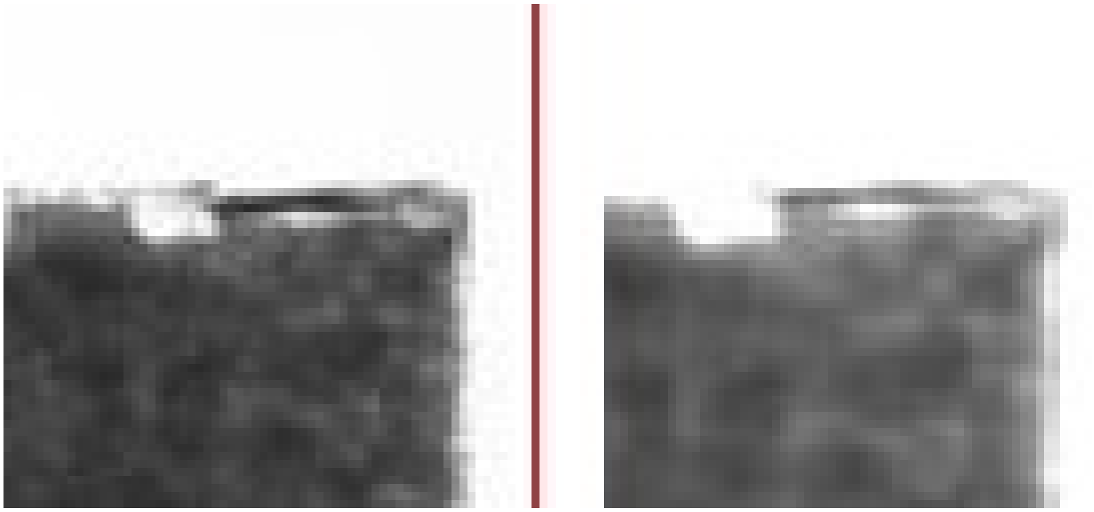


Figure 2.4: A side-by-side comparison of a corner of a MIP film before, on the left, and after, on the right, filtering is applied.

2.2.2 Intensity Gradients

The intensity gradients, which is the rate at which the pixel intensity changes in a certain direction over some distance, is then calculated for the filtered image. The intensity gradients are calculated for the horizontal, and vertical directions, since an edge can point in either direction. The operator is a discrete differentiation operator which approximates the gradient of the images intensity function. The operator uses two kernels that are applied to the image's subsections through a convolution operation as follows

$$\mathbf{M}_x = \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} * \mathbf{I}' \tag{2.5}$$

$$\mathbf{M}_y = \begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix} * \mathbf{I}' \tag{2.6}$$

Above, $\mathbf{M}_x$ is the gray-scale image with intensity at each point calculated from the horizontal derivative of $\mathbf{I}'$, the original gray-scale image, at that same point. Similarly, $\mathbf{M}_y$ is an image constructed of pixel values which are the vertical derivatives of $\mathbf{I}'$. The resulting gradient magnitude, $\mathbf{M}$, and direction, $\boldsymbol{\Theta}$, are then calculated using

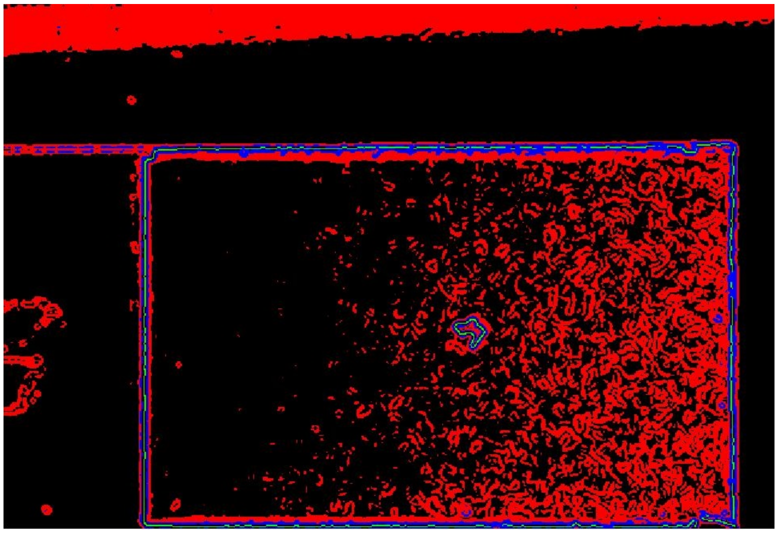
$$\boldsymbol{\Theta} = \arctan\left(\mathbf{M}_y, \mathbf{M}_x\right) \tag{2.7}$$

and

$$\mathbf{M} = \sqrt{\mathbf{M}_x^2 + \mathbf{M}_y^2} \tag{2.8}$$

which contain the gradient magnitude and direction for each point in the image. The result of applying this operation to the image, in which each pixel's intensity is given by $\mathbf{M}$, is shown in Fig. 2.5a.

(a) The image with pixels given by the gradient of the image in Fig. 2.3b.



(b) A color coded version of the gradient image of a MIP film shown in Fig. 2.5a. Black pixels have intensity between 0 and 0.05, red pixels have intensity between 0.05 and 0.2369, blue pixels have intensity between 0.2369 and 0.5922, and green pixels have intensity higher than 0.5922.

Figure 2.5: The gradient of a MIP film shown in Fig. 2.3b.

2.2.3 Non-Maximum Suppression

Using the gradient image obtained in Sec. 2.2.2, non-maximum suppression is applied. Non-maximum suppression is an edge thinning technique which seeks to remove points within an edge that are either noise or are superfluous. The goal of non-maximum suppression is to set every gradient to zero which is less than the local maximum where the most prominent edge points occur. The algorithm for non-maximum suppression is as follows:

1. Iterate over each pixel in the intensity gradient image
2. Compare the intensity of pixel i, j with the intensity in the positive and negative gradient directions
3. If the intensity is the local maximum, preserve it
4. Else suppress it

The positive gradient direction is given by Eqn. 2.7. The result of this step is depicted in Fig. 2.6, which shows a logical matrix (where each pixel is either 1 or 0). Each white pixel indicates that the pixel is a local maximum in the gradient image shown in Fig. 2.5a.

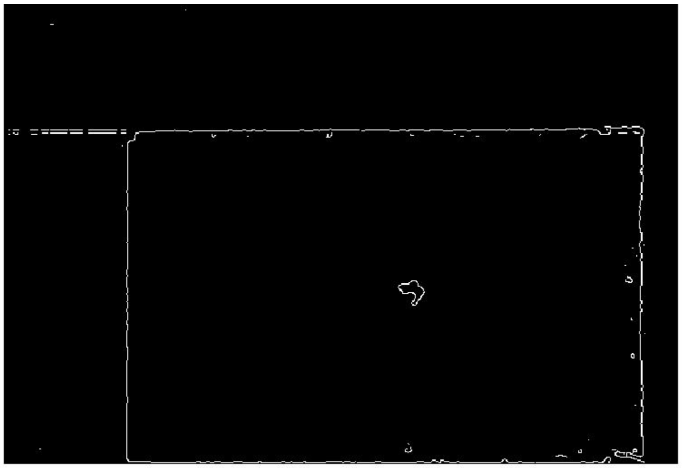


Figure 2.6: An image of a MIP in the non-maximum suppression step in the Canny edge detection algorithm, where each white pixel indicates the corresponding pixel is a local maximum in the gradient image in Fig. 2.5a

2.2.4 Double Threshold

Some edge pixels remain in Fig. 2.6 due to noise and color variation, even after applying the first several steps of the Canny edge detection process. Thus, weak gradient pixels are filtered out using a double threshold step. In this step, two threshold values are determined either empirically or by using an algorithm. In this thesis, the upper threshold was chosen based on inspection of the distribution of the magnitude of gradients which were obtained from images such as the one in Fig. 2.5a. The actual distribution is shown in Fig. 2.7. The upper threshold was set to where the distribution began to decline at the end of the plateau. The lower threshold was manually set to be 0.4 times the upper threshold, which coincided roughly with where the distribution tapered off and plateaued. These thresholds were used for all images within the data set.

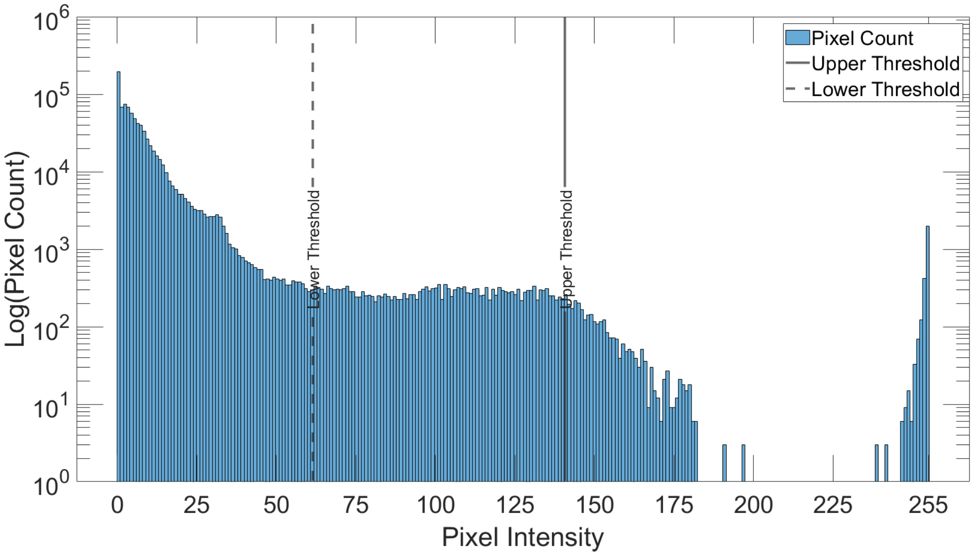
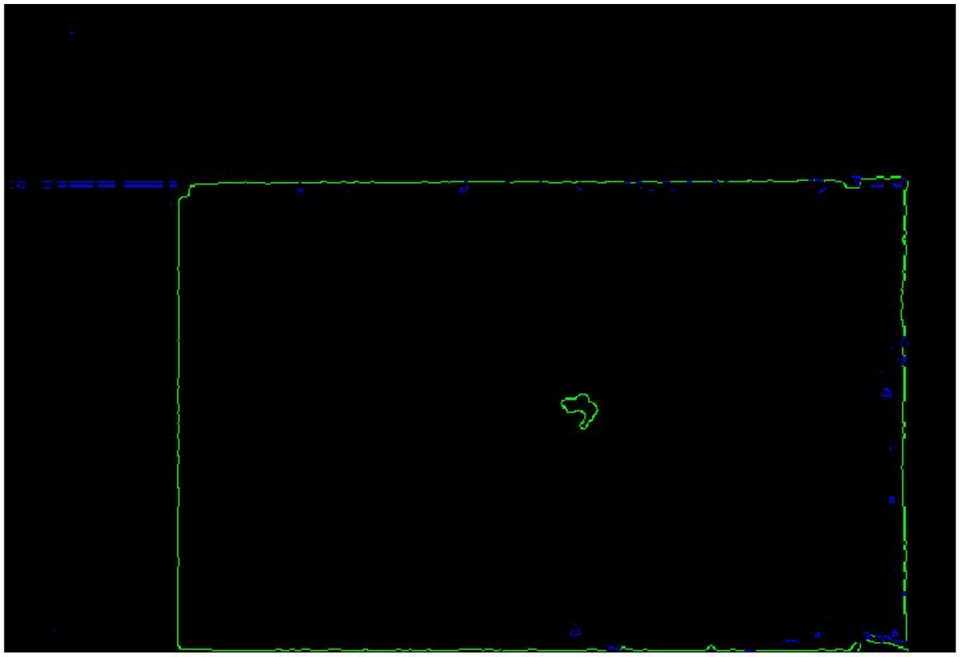


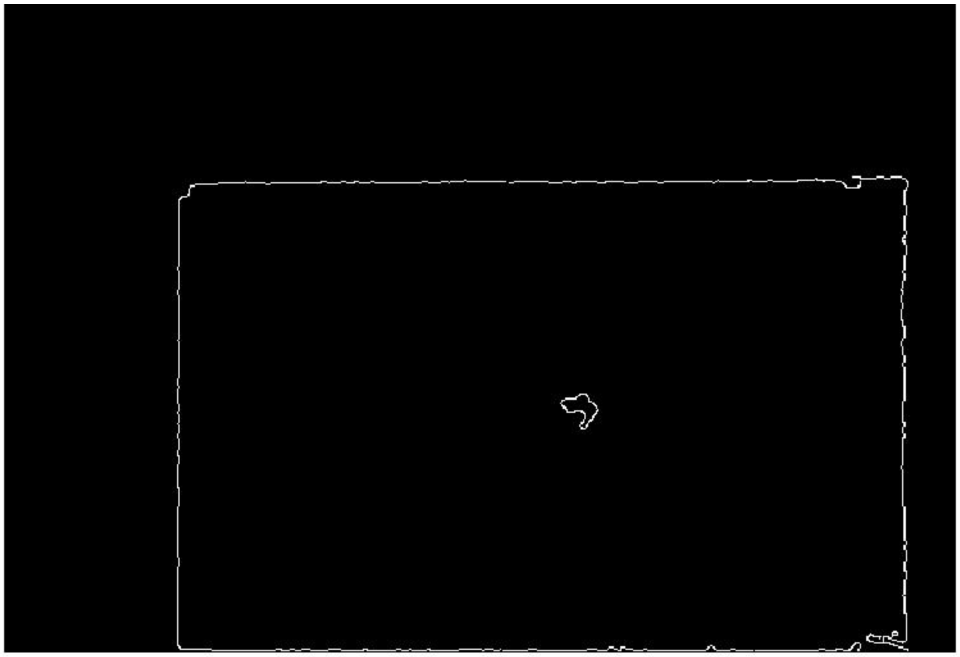
Figure 2.7: The log of the distribution of the magnitudes of the gradients from the image shown in Fig. 2.5a, where the solid and dashed lines indicate the manually chosen upper and lower thresholds, respectively, used for the double threshold step.

2.2.5 Hysteresis-Based Edge Tracking

Weak pixels discovered in the step described in Sec. 2.2.4 require further work to determine whether they will be preserved or suppressed, while strong pixels are automatically included in the final Canny edge image. Weak pixels that are connected to a blob, which is a collection of pixels that are connected through horizontal, diagonal, or vertical connections, that contain at least one strong pixel are preserved while the rest are suppressed. In Fig. 2.8a, a double threshold was used to classify each gradient pixel in Fig. 2.6. Blue pixels are weak, since they fall between the two thresholds, and any that are not connected to a strong, green pixel are removed from the image. Green pixels are strong, since they fall above both thresholds, and are preserved. The final result is shown in Fig. 2.8b.



(a) A color coded version of the local maximum gradient image of a MIP film shown in Fig. 2.6. Green pixels are strong, and blue pixels are weak according to the selected double threshold.



(b) A MIP film after being processed in the hysteresis-based edge tracking step in the Canny edge detection algorithm, where each white pixel indicates a final edge pixel.

Figure 2.8: The final Canny edge image of the MIP film shown in Fig. 2.3a.

2.3 Hough Transformations

Hough transformations [32] are applied to the Canny edge images obtained in the previous step to find the presence of lines in a process facilitated by using a homogeneous coordinate system to represent lines. A valid line is one that passes through many Canny edge points. ρ is the perpendicular distance between the line and the origin and θ gives the direction of the line, as shown in Fig. 2.9.

2.3.1 Homogeneous Coordinate Systems

Lines in the xy plane can be represented using homogeneous coordinates denoted by $\mathbf{l} = (a, b, c)$ [32]. The equation of a line in this coordinate system is given by

$$\mathbf{x} \cdot \mathbf{l} = ax + by + c = 0 \tag{2.9}$$

Eq. 2.9 can be normalized so that $\mathbf{l} = (\hat{\mathbf{n}}_x, \hat{\mathbf{n}}_y, \rho)$, where $\hat{\mathbf{n}}$ is the normal vector perpendicular to the line and ρ is its distance to the origin. This normal vector can be expressed in terms of the polar coordinate, θ, and is written as $(\cos\theta, \sin\theta)$. Thus, using this homogeneous coordinate system a straight line can be represented in terms of coordinates θ and ρ. The straight line is expressed as

$$\rho = x\cos\theta + y\sin\theta \tag{2.10}$$

from which the conventional equation of the line

$$y = mx + b \tag{2.11}$$

may be derived, where m and b can be written in terms of ρ and θ as

$$m = -\frac{\cos\theta}{\sin\theta} \tag{2.12}$$

and

$$b = \frac{\rho}{\sin\theta}. \tag{2.13}$$

Such a line is shown in Fig. 2.9.

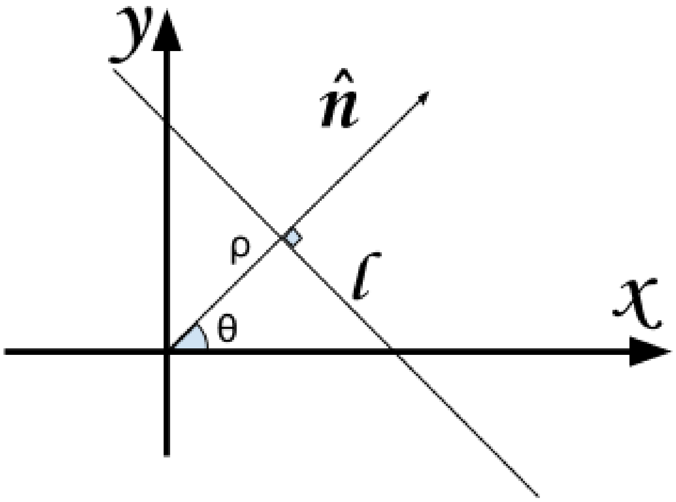


Figure 2.9: Depiction of a straight line characterized by parameters ρ and θ in a Cartesian coordinate system.

2.3.2 Applying a Hough Transform

A Hough transformation is a process by which voting occurs to determine the presence of plausible straight edges [32]. The result of a Hough transformation is the creation of the so-called accumulator space of the image, which is a three-dimensional space spanned by vectors represented by θ, ρ (as in Sec. 2.3.1), and the number of votes, C, for that line. The accumulator space is constructed by creating various combinations of θ and ρ, and counting the number of pixels intersected by the corresponding line, giving the number of votes. Straight edges are distinguished by having a high number of votes in the accumulator space. For example, a line that intersects a long row of points, i.e. an edge, will have a high number of votes and therefore will be represented in the accumulator space as a peak.

2.3.3 Sorting Lines

Straight edges in the image that are obtained from the Canny edge detection algorithm appear as peaks when processed with a Hough transform. Each of the peaks' lines are characterized by parameters θ and ρ and Eqn. 2.10. To increase computational efficiency, all chosen peaks are organized into two sets: one containing lines that are considered vertical, and another containing lines that are considered horizontal.

The logic that distinguishes horizontal lines from vertical lines is given by

$$\begin{gathered}\text{Horizontal if } 45^\circ < \theta \leq 135^\circ \text{ or } 225^\circ \leq \theta \leq 315^\circ \\ \text{Vertical if } -45^\circ \leq \theta \leq 45^\circ \text{ or } 135^\circ \leq \theta \leq 225^\circ\end{gathered}$$

The lines are sorted in order to reduce the number of combinations of lines used to find the four lines corresponding to the sides of the film. This also helps with finding which corner of the MIP corresponds to the intersection of two lines.

2.4 Image Pre-Processing

Prior to using the pipeline, the image is processed such that extraneous parts of the image are removed. This involves removing the lower part of the image which includes the ruler/support for the MIP. Since the ruler is in the exact same place in every picture, the equation of the line of the top of ruler may easily be found by manually picking the points at the left and right end of the ruler and determining its slope. From there, one may easily find the θ and ρ coordinates of the line. The θ coordinate is used to rotate the image, and the ρ coordinate is used to determine up to what height of the image is kept in the resulting image shown in Fig. 2.10.

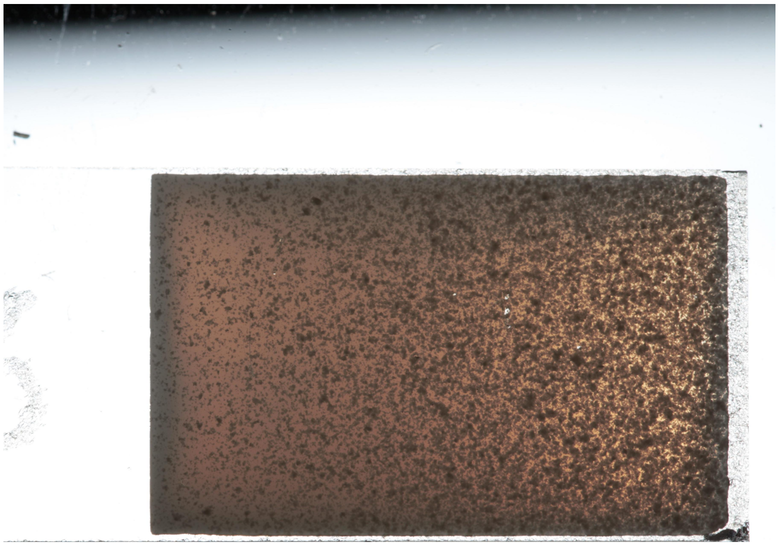


Figure 2.10: An image of a MIP film which has had its lower portion removed and the image rotated.

2.5 Detection of the MIP-containing Region

The MIP film is first detected to determine the geometry of the MIP film and then isolated to ensure only MIP film textures are used for subsequent machine-learning related steps. This is achieved by finding a set of four lines which forms the rectangle that coincides with the film. As discussed in Sec. 2.3.3, the set of all candidate lines are sorted into two sets; one containing horizontal lines and the other vertical lines. To find the four lines which correspond to the MIP film, two are chosen from each set. Once the four lines are chosen, the points at which these four lines intersect are taken as the four vertices of the MIP film. The procedure used to find the film involves finding sets of straight lines present within the image and finding the set that satisfies

$$\underset{\alpha\beta\gamma}{\arg\max}\{g = w_\alpha \alpha + w_\beta \beta + w_\gamma \sum_i^4 \gamma_i\} \tag{2.14}$$

where α is the score of the surface area, β is the score of the aspect ratio, γ is the score of the internal angles, and the coefficients are simply weights used to signify the importance of each characteristic's score. The functions that determine each of the scores are constructed from continuous piece-wise linear functions which have a peak at the most desirable value. The justification for this procedure is that MIP films are more or less constrained to a range of values for surface area, aspect ratio, and internal angles. Secondly, there exist few other candidate rectangles within each image which could possibly outscore the MIP film. Thus, it is assumed that the best scoring rectangle is the true MIP film rectangle. Finding the area that contains the film allows the system to obtain a measurement of the geometric properties of the MIP. It also allows the system to remove all pixels outside the selected region which reduces the size of the data set for subsequent steps of the pipeline. Below, we describe the equations used to score a given rectangle.

The score of the surface area, indicated by A, is determined by measuring the internal area of the polygon given by the four points of intersection of the four lines under consideration.

$$\alpha = \begin{cases} 0 & \text{if } A < 170\text{mm}^2 \\ \frac{1-0}{188.5-170} \cdot (A - 170) & \text{if } 170\text{mm}^2 < A <= 188.5\text{mm}^2 \\ \frac{0-1}{207-188.5} \cdot (A - 207) & \text{if } 188.5\text{mm}^2 < A <= 207\text{mm}^2 \\ 0 & \text{if } A > 207\text{mm}^2 \end{cases}$$

The score of the aspect ratio, indicated by R, is calculated by finding the four pairs of intersecting lines and dividing the longer side's length by the shorter side's length. These four resulting aspect ratios are then averaged to obtain R.

$$\beta = \begin{cases} 0 & \text{if } R < 1.41 \\ \frac{1-0}{1.61-1.41} \cdot (R - 1.41) & \text{if } 1.41 < R <= 1.61 \\ \frac{0-1}{1.81-1.61} \cdot (R - 1.81) & \text{if } 1.61 < R <= 1.81 \\ 0 & \text{if } R > 1.81 \end{cases}$$

The score of the internal angles is calculated by finding each of the four internal angles, denoted by I, assigning to each a sub-score between 0 and 0.25, and adding them up to obtain an overall score between 0 and 1.

$$\gamma_i = \begin{cases} 0 & \text{if } I < 88 \\ \frac{0.25-0}{90-88} \cdot (I - 88) & \text{if } 88 < I <= 90 \\ \frac{0-0.25}{92-90} \cdot (I - 92) & \text{if } 90 < I <= 92 \\ 0 & \text{if } I > 92 \end{cases}$$

The values of the weight coefficients are given by:

$$w_\alpha = 2 \tag{2.15}$$

$$w_\beta = 1 \tag{2.16}$$

$$w_\gamma = 3 \tag{2.17}$$

It should be noted that these particular sets of characteristic value ranges and the weight coefficients are empirically determined for the given set of MIP films. They are intended to be set and tuned in consultation with the user of the quality control system.

2.6 Edge Trimming

Since the process described in Sec. 2.5 may not always obtain the four lines corresponding to the four sides of the film, due to the presence of a rectangular glass slide which can also provide highly scoring candidate lines, an additional "edge-trimming" step is used. Consider the following MIP which has two of the four selected lines erroneously chosen due to the presence of the glass slide and excess solvent from the manufacturing process.

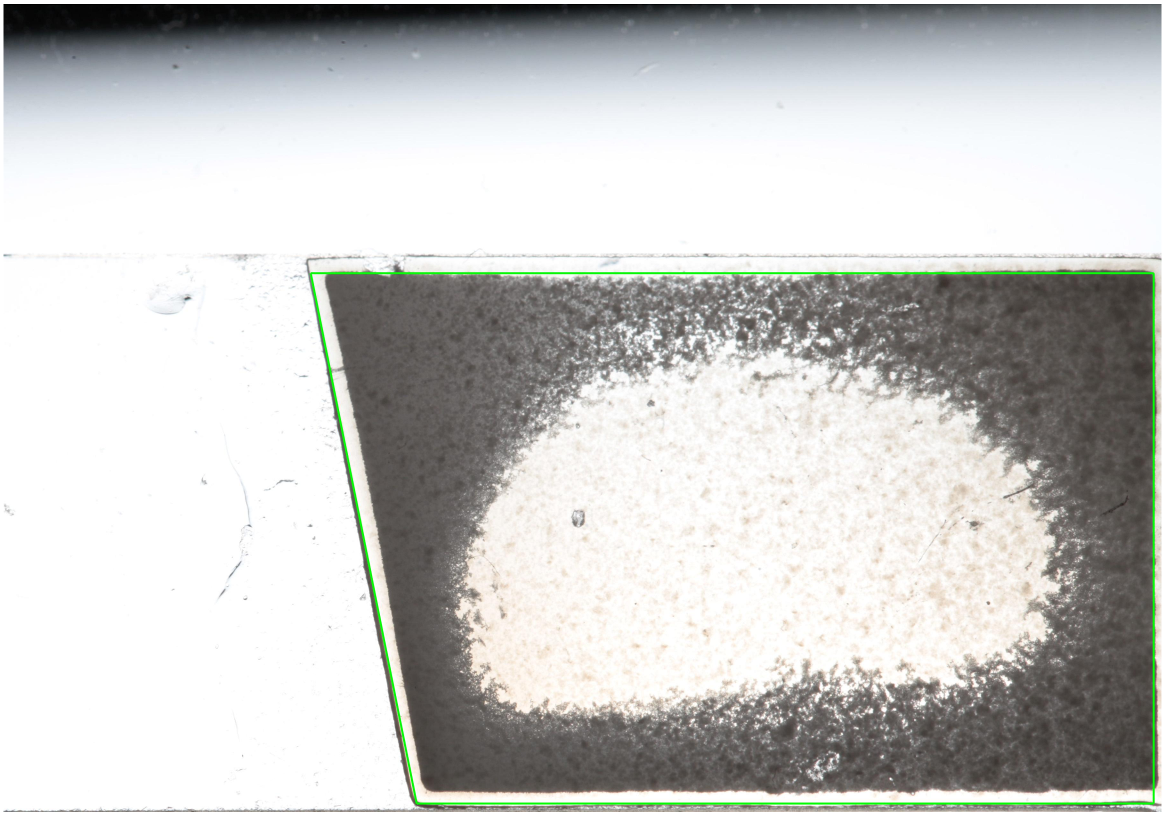


Figure 2.11: An example defective MIP film which has had its four lines incorrectly selected.

Edge trimming involves gradually shifting each of the selected lines until a sufficiently dark region is reached. Each line is shifted by increasing or decreasing the ρ

component of the line. A binary image is constructed from this image and the total number of white pixels within the enclosed region, shown in Fig. 2.12, is counted. If it is higher than a chosen threshold, the shifted line replaces the original line. This process repeats until the number of enclosed white pixels no longer surpasses the minimum desired level.

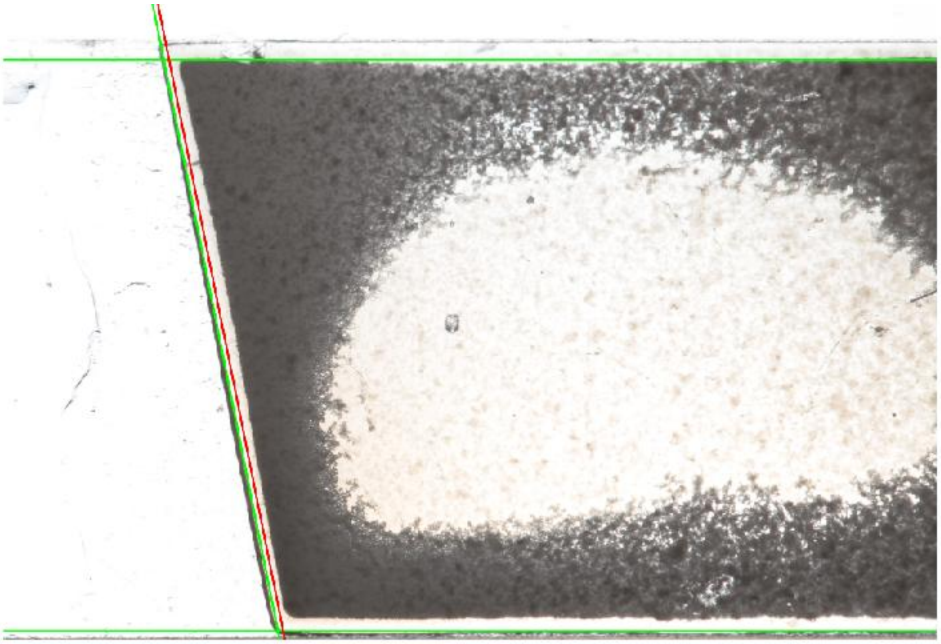


Figure 2.12: A MIP film, in the process of being trimmed, with its initially selected lines shown in green and the partially trimmed line shown in red which will replace its parent line in the subsequent step of the line trimming algorithm.

2.7 Calculation of MIP Side Lengths

The dimensions of the MIP films are of interest in quality control as surface area affects both performance and a customer's perception of the film's quality. In order to calculate the side lengths, one must find the points at which adjacent sides intersect. Using a homogeneous coordinate system as described in Sec.2.3.1, the points at which two lines intersect is given by [32]

$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2 \tag{2.18}$$

Since the lines have already been sorted into two sets, one for “vertical” lines and the other for “horizontal” lines (meaning they are not necessarily parallel to the y or x axis respectively), one need not worry about the possibility of mistakenly finding the point of intersection between two vertical lines, or two horizontal lines. The four points of intersection of interest are calculated by finding all pairs of lines by taking one line from each set. For example, in Fig. 2.13 the red and blue sides are known to be vertical, and the yellow and green sides are known to be horizontal. The pairs of lines of interest are found by taking a Cartesian product of these two sets.

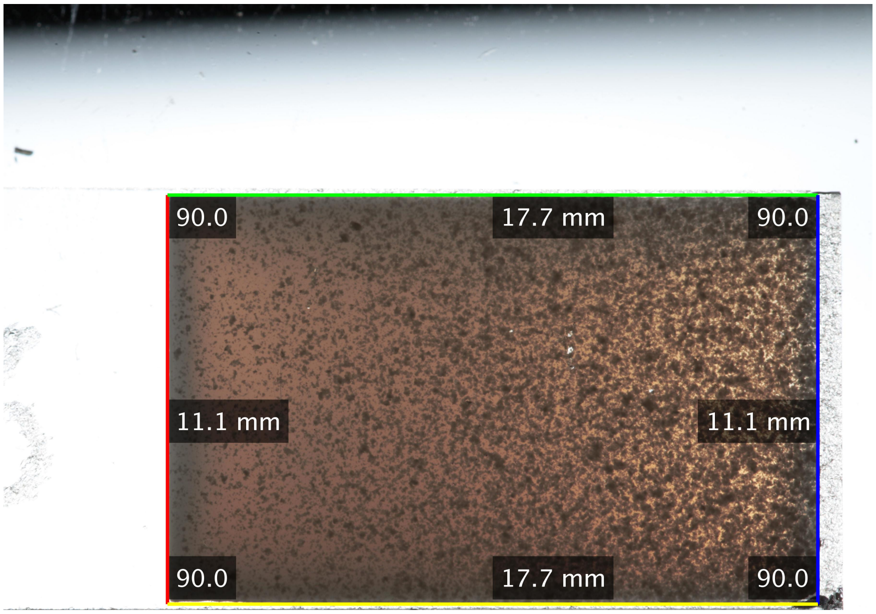


Figure 2.13: A MIP film for which the geometry characteristics are overlaid on the image

Once all four points of intersection are found, each point is paired with its two adjacent points. A vector is created from each pair, and the length of the vector using a Euclidean norm is calculated to give the length of the film side. The result is depicted in Fig. 2.13.

2.8 Rotation of the MIP

A MIP film may be oriented at an angle relative to the bottom of the image frame, even after the trimming and leveling step described in Sec. 2.4 since the MIP film itself is not guaranteed to be leveled with the ruler. Thus, the image is rotated again to standardize the input data and make extracting the interior of the film easier. Consider Fig. 2.14 in which the θ value for each line is represented by a colored, circular segment.

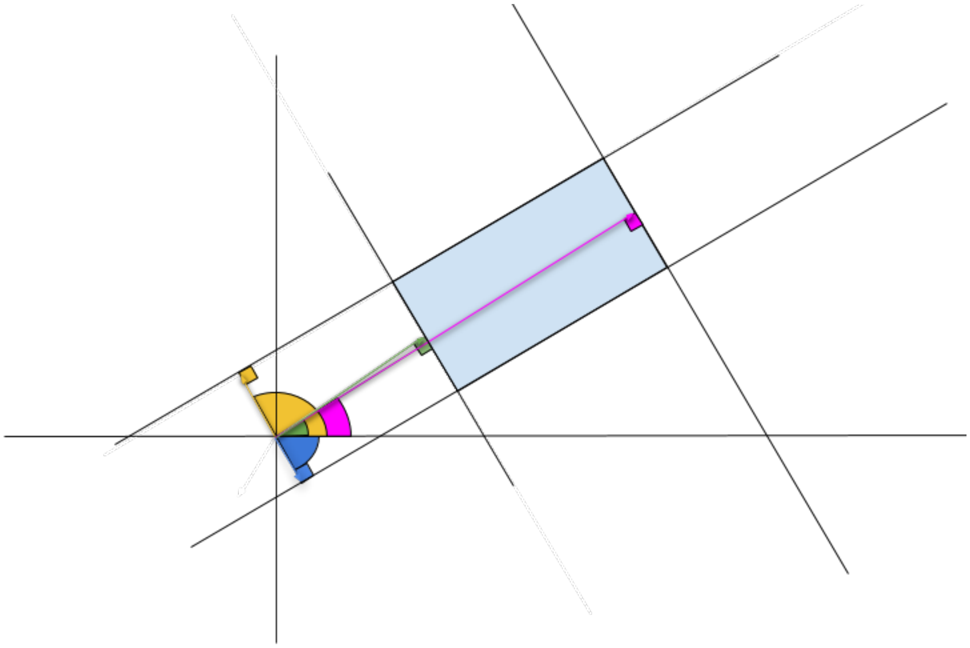


Figure 2.14: An illustration depicting the lines and their corresponding θ values, obtained from a Hough transform, for a MIP film.

Each normal line is colored, and its corresponding θ value is colored the same. The θ values can be used to find the rotation angle that levels the rectangle in this coordinate system. Through inspection, this rotation angle is

$$-\frac{1}{4}\sum_{i=1}^{4}\mathrm{mod}(\theta_i, 90) \tag{2.19}$$

which is the negative of the average of each line's angular distance to its closest axis (in the clock-wise direction). It is then added to each of the four lines' θ value to set the lines so they become parallel to either the y or x axis. A built-in MATLAB function for rotating the actual images is used with this angle to rotate the actual image.

2.9 Edge Quality

The quality of the edge of a MIP film can be quantified in two ways: regularity, and strength. Edge regularity is a measure of how close the Canny edge is to the Hough transform line on average. If the Canny edge coincides perfectly with the line, it is perfectly regular. If it oscillates above and below, or runs parallel to the Hough transform line with a large separating distance, the Canny edge is considered very irregular. Edge strength is a measure of how many edge pixels exist in a given area of an image. In other words, it is a measure of edge pixel density. The more Canny edge pixels per unit of area, the higher the strength. Inversely, a low edge pixel density means the edge is weak for the area of the image under inspection. Both of these properties are calculated for segments of the sides of the MIP film to obtain local edge strength and regularity measures.

First, an area is defined for a given line segment on the Hough transform line of length L. For each line segment, a search area of length L and width W is formed, centered on the Hough transform line. The direction in which the length is defined is always parallel with the line segment under consideration. Edge regularity is obtained by calculating the average distance of the points in the Canny edge, within the search area, to the line found by the Hough transform. In this thesis, $L = 0.8\%$ of the image height, and $W = 0.4\%$ of the image width, which were approximately 30 and 20, respectively. An example of a search area is shown in Fig. 2.15.

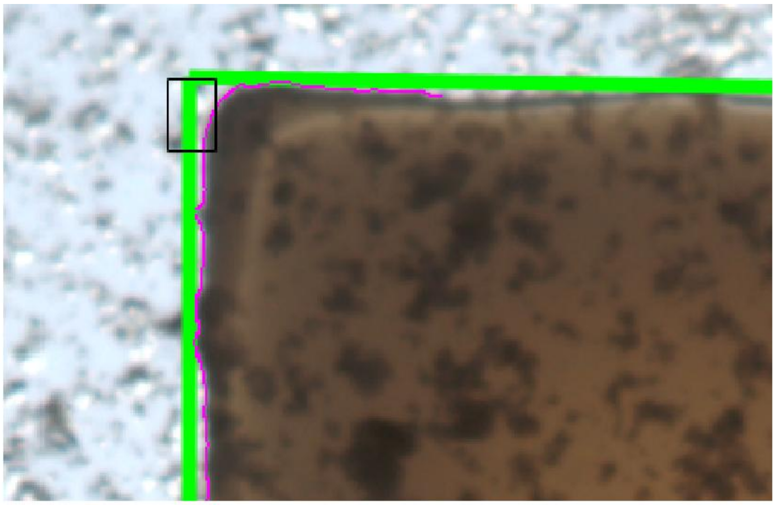


Figure 2.15: The lines obtained from the Hough transform are shown in green, the edge obtained from the Canny edge detector is shown in pink, and the search area to determine edge quality is shown in black.

The reasoning behind this step is that if there are any edge pixels that exist in close proximity to the Hough line but do not coincide with it, it must mean the edge is irregular and is therefore undesirable. Thus, a greater edge irregularity implies a lower quality MIP film. The height of the box is defined so that only pixels that are reasonably assumed to belong to the MIP film are chosen to calculate edge roughness. If the height is too large, then the algorithm will incorporate edge pixels from, say, the edge of the glass slide, which will result in a large average deviation from the Hough line, skewing the regularity measure. The regularity measure is defined as

$$E = \frac{\sum_i^N d(P_a, P_b, (x_i, y_i))}{N} \tag{2.20}$$

which could equivalently be called "Mean Absolute Error," where N is the number of edge points in the search area, P_a is the beginning point of the line segment of the Hough transform line, P_b is the endpoint of the line segment, (x_i, y_i) is the coordinates of the ith Canny edge pixel, and the function d is given by

$$d(P_a, P_b, (x_0, y_0)) = \frac{|(y_b - y_a)x_0 - (x_b - x_a)y_0 + x_b y_a - y_b x_a|}{\sqrt{(y_b - y_a)^2 + (x_b - x_a)^2}}, \tag{2.21}$$

which gives the perpendicular distance of a Canny edge pixel to the line formed by P_a and P_b.

Clearly, the more the Canny edge points deviate from the lines found from the Hough transform the higher the error will be. A line segment is classified as regular if it has an error of less than $0.15W$. A line segment is classified as weak if it has fewer than $\frac{LW}{40}$ Canny edge pixels. An example of a weak edge may also be seen in Fig. 2.15 at the top of the MIP film where there are evidently no Canny edge pixels.

2.10 Internal Angles

One property of interest for evaluating the quality of MIPs is the set of a MIP film's internal angles. Depending on manufacturing feasibility and marketability, MIPs must be produced consistently with specific internal angles. Thus, internal angles should be used to assess quality. The angle of intersection between two sides is obtained through forming a vector from each line, by using the line's endpoints, and calculating the angle using

$$\phi = \arccos \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \tag{2.22}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the vectors corresponding to each of the two lines.

2.11 Threshold-Based Defect Detection

An additional simpler technique used to find defects is based on applying a high threshold to the images of the MIP films to form a black and white image, where the values of pixels are either one or zero, and calculating the total number of white pixels in the image. The premise is simple: due to the underlying white light shining through the glass slide any scrapes should appear in black and white images for high thresholds while all other pixels become suppressed, which can be seen in Fig. 2.16 and Fig. 2.17. The purpose of including this method is to compare its defect classification accuracy with the classification accuracy of the more complex SOM to determine the SOM's feasibility as a defect detection algorithm.

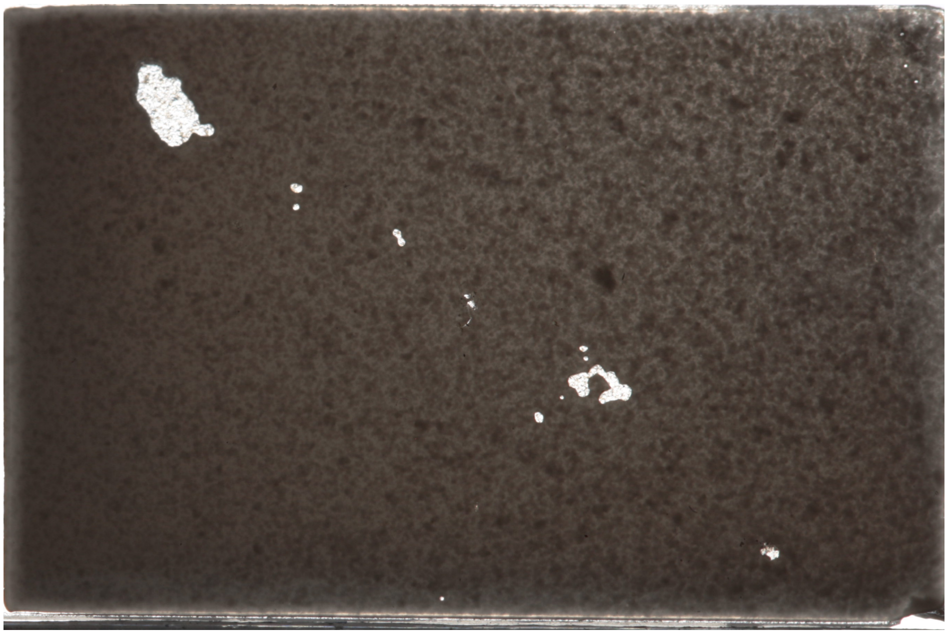


Figure 2.16: A defective MIP film which has been isolated from its background.

Figure 2.17: A black and white image of Fig. 2.16 with threshold 0.75.

2.12 Calculation of Feature Vector

Each image of a MIP film is divided into cells, and a feature vector is calculated for each cell. Each feature vector consists of features which describe their corresponding image cell. The raw feature vectors used for the data set are constructed from twenty different features. Features 1 and 2 are simple statistical measures, features 3-14 are Haralick texture features, and features 15-20 are custom features. The features used are as follows:

1. Mean gray Level
2. Median gray level
3. Angular second moment
4. Contrast
5. Variance
6. Inverse difference moment
7. Sum average
8. Sum variance
9. Sum entropy
10. Entropy
11. Difference variance
12. Difference entropy
13. Information measure of correlation II
14. Maximal correlation coefficient
15. Blob, threshold 0.5
16. Blob, threshold 0.6
17. Blob, threshold 0.7

18. Blob, threshold 0.8
19. Blob, threshold 0.9
20. Truth transition distance

2.12.1 Gray Levels

Mean gray level is calculated by first converting the image to a gray-scale image. In MATLAB, this is performed by calling "rgb2gray." The intensity values, between 0 and 255, are then added to a running total for each pixel in the image and then averaged over the total number of pixels in the image. Similarly, the median gray level is calculated by finding the median intensity value for all pixels.

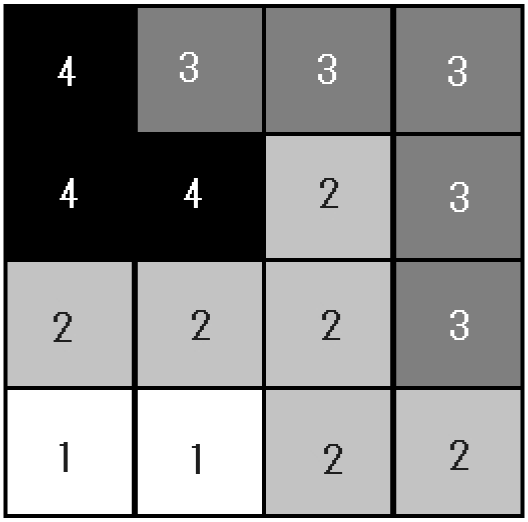
2.13 Haralick Texture Features

Image texture describes how the color/intensity levels in an image vary from pixel to pixel. According to Haralick [33][34], image texture is described by the number and types of so-called "tonal primitives" and the spatial organization of those primitives. Tonal primitives are maximally connected regions of an image which share the same tonal properties. A tonal property is described by gray tone properties, such as the highest or lowest gray level, and the tonal region properties, such as the shape or size. Thus, image texture is described by the number and types of its primitives and spatial organization of its primitives. Haralick captures the texture information from images composed of tonal primitives using the Haralick texture features.

2.13.1 Gray Level Co-Occurrence Matrix

The Haralick texture features describe the spatial inter-relationships of gray tones in textural patterns by making use of a so-called gray level co-occurrence matrix (GLCM). For each image cell in this thesis, the Haralick texture features are computed and therefore a GLCM is also computed for each cell. The GLCM can be specified by a matrix of relative frequencies p. Each element of p, $p(i, j)$, is the frequency at which a pixel with gray level j occurred next to a pixel with gray level i in a specified direction and separated by distance d. GLCMs can be constructed by finding neighboring pixels in any direction; vertically, horizontally, diagonal, or a combination of the three. This co-occurrence approach characterizes the spatial distribution and spatial dependence among the gray tones in a local area. To elaborate on the meaning of the GLCM, consider the example image shown in Fig. 2.18. The GLCM for this image was created using an offset in the positive x-direction with a distance of one and is given by

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$



4	3	3	3
4	4	2	3
2	2	2	3
1	1	2	2

Figure 2.18: A simple 4×4 image with 4 possible gray levels.

The GLCM can be made into a symmetric matrix by instead adding up the co-occurrence of pixel intensities in both positive and negative directions of the offset which generates the resulting GLCM of

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 6 & 2 & 1 \\ 0 & 2 & 4 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

Computing a symmetric GLCM assists in efficient computation of the Haralick texture features since it allows one to reduce the number of calculations by considering the symmetry of the matrix. In this thesis, GLCMs are calculated by using an offset in the positive and negative x-direction with a distance of 1. Only this direction is used for speeding up processing since additional offsets require computing additional GLCMs and therefore increases computing time. However, it should be noted that one benefit of using multiple GLCMs is that it facilitates insensitivity to rotation or scale of the images.

The Haralick features are calculated as follows [35]. G is equal to 256, the number of gray levels used, and μ is the mean of p.

$$\text{Angular Second Moment (ASM)} = \sum_i \sum_j p(i,j)^2 \tag{2.23}$$

$$\text{Contrast} = \sum_{n=0}^{G-1} n^2 \sum_i^G \sum_j^G p(i,j), |i-j| = n \tag{2.24}$$

$$\text{Variance} = \sum_i \sum_j (i-\mu)^2 p(i,j) \tag{2.25}$$

$$\text{Inverse difference moment (Homogeneity)} = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j) \tag{2.26}$$

$$\text{Sum average} = \sum_{i=2}^{2G} i p_{x+y}(i) \tag{2.27}$$

$$\text{Sum entropy} = -\sum_{i=2}^{2G} p_{x+y}(i) \log\left(p_{x+y}(i)\right) = f_s \tag{2.28}$$

$$\text{Sum variance} = \sum_{i=2}^{2G} (i-f_s)^2 p_{x+y}(i) \tag{2.29}$$

$$\text{Entropy} = -\sum_i \sum_j p(i,j) \log\left(p(i,j)\right) \tag{2.30}$$

$$\text{Difference Variance} = \text{Variance of } p_{x-y} \tag{2.31}$$

$$\text{Difference Entropy} = -\sum_{i=0}^{G-1} p_{x-y}(i) \log\left(p_{x-y}(i)\right) \tag{2.32}$$

$$\text{Information measure of correlation II} = (1 - \exp^{-2(HXY2-HXY)})^{1/2} \tag{2.33}$$

$$HXY = -\sum_{i}\sum_{j} p(i,j) \log\left(p_x(i)p_y(j)\right) \tag{2.34}$$

$$HXY2 = -\sum_{i}\sum_{j} p_x(i)p_y(j) \log\left(p_x(i)p_y(j)\right) \tag{2.35}$$

$$\text{Maximal Correlation Coefficient} = \text{Square root of the second largest eigenvalue of Q} \tag{2.36}$$

where

$$Q(i,j) = \sum_{k} \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)} \tag{2.37}$$

and $p_x(i)$, $p_y(j)$, $p_{x+y}(i)$, and $p_{x-y}(i)$ are given by

$$p_x(i) = \sum_{j=1}^{G} p(i,j) \tag{2.38}$$

$$p_y(j) = \sum_{i=1}^{G} p(i,j) \tag{2.39}$$

$$p_{x+y}(k) = \sum_{i=1}^{G} p(i,j), i+j = k \text{ and } k = 2, 3, ..., 2G \tag{2.40}$$

$$p_{x-y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} p(i,j), i - j = k \text{ and } k = 0, 1, ..., G-1 \tag{2.41}$$

$p_x(i)$ is the sum of the ith row of p, and $p_y(j)$ is the sum of the jth column of p. These essentially give the number of times a pixel with intensity i (or j) occurred next to a pixel of any intensity. We refer the reader to the original paper by Haralick and coworkers [34] for an exposition on how to intuitively understand these texture features. Note that the correlation feature is not used in this thesis, since it becomes undefined for certain images.

2.14 Calculation of Intensity Resilience

The calculation of intensity resilience is a measure of how much the intensity of the image changes with respect to increasing threshold. An image with homogeneous, low gray levels will darken quickly with respect to an increasing threshold. An image with one or more bright spots will maintain such bright spots for higher thresholds than dimmer, homogeneous images. Here we introduce the concept of intensity transition distance, later called "truth transition distance" (TTD).

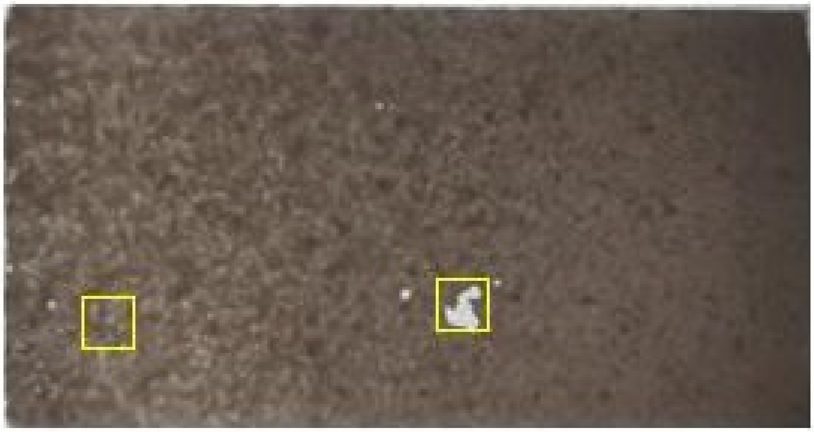


Figure 2.19: An example of a MIP film with two regions of different texture which is highlighted.

Consider Fig. 2.19. Two regions are highlighted: the one on the left containing light speckles and the other containing a very visible scratch in which the underlying

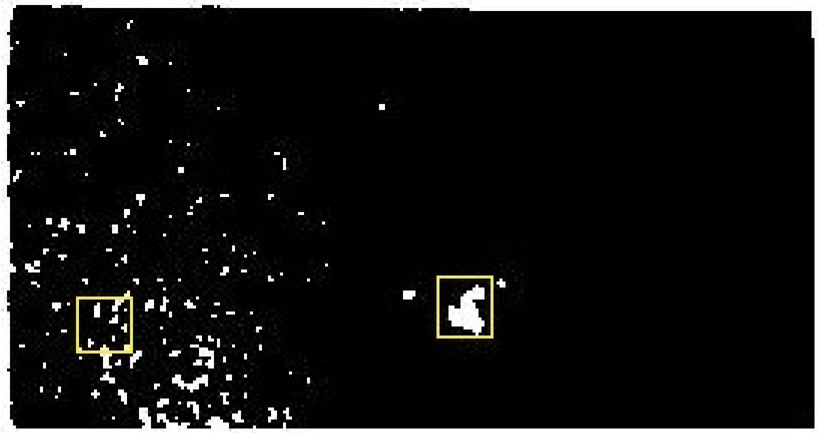


Figure 2.20: An binary image of the MIP film shown in Fig. 2.19

illuminated glass slide is visible. The image is converted to a black and white binary image shown in Fig. 2.20.

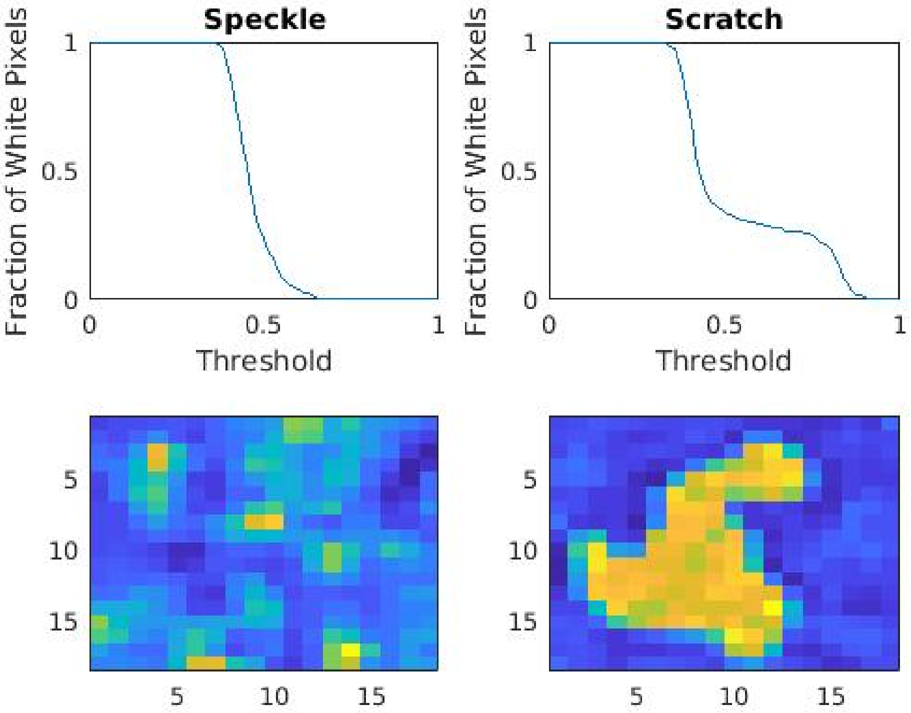


Figure 2.21: Plots, and their corresponding image cells, which show how the total fraction of the image that is white decreases with respect to increasing threshold.

In Fig. 2.21, the lower left image is a closeup of the prominent scratch in Fig. 2.19 while the lower right image is a closeup of the speckled region. The plots above are the total fraction of the highlighted image sections that are made up of white pixels with respect to increasing threshold. For each threshold value, the image is converted to a black and white image with the threshold applied. For the scratched regions, the fraction of white pixels, or "truth" as we call it, remains higher for higher thresholds compared to speckled regions. This brings us to the concept of "truth transition distance," or how long it takes for an intensity image to completely darken with respect to increasing threshold. According to Fig. 2.21, this quantity, which is just the arc length of the curve for truth values greater than 0.0001, can therefore be used to differentiate bright scratches from dull, dimly lit, or speckled regions. Additionally, the features referred to as "blob" are calculated by finding the largest maximally-connected group of white pixels within the image cell for a given threshold, calculating its size, and normalizing it by the surface area of the image cell it is contained within. For example, "blob 0.5" finds the blob value at threshold=0.5.

2.15 Principal Component Analysis

PCA is an orthogonal linear transformation which maps a set of vectors defined in some space to a new space spanned by orthogonal basis vectors such that the greatest variance of the transformed data set occurs along the first basis vector, and the second greatest variance of the data occurs along the second, and so on [36]. Consider Fig. 2.22.

Figure 2.22: A two-dimensional data set plotted with respect to its two feature variables.

Variable x_1 is clearly highly correlated with variable x_2. Upon observation, it is also clear that there seems to be a single direction in which the data vary the most. This direction can be obtained using PCA, which will take an n-dimensional data set of m observations and return n principal component vectors. In this case, since $n = 2$, there will be two principal component vectors.

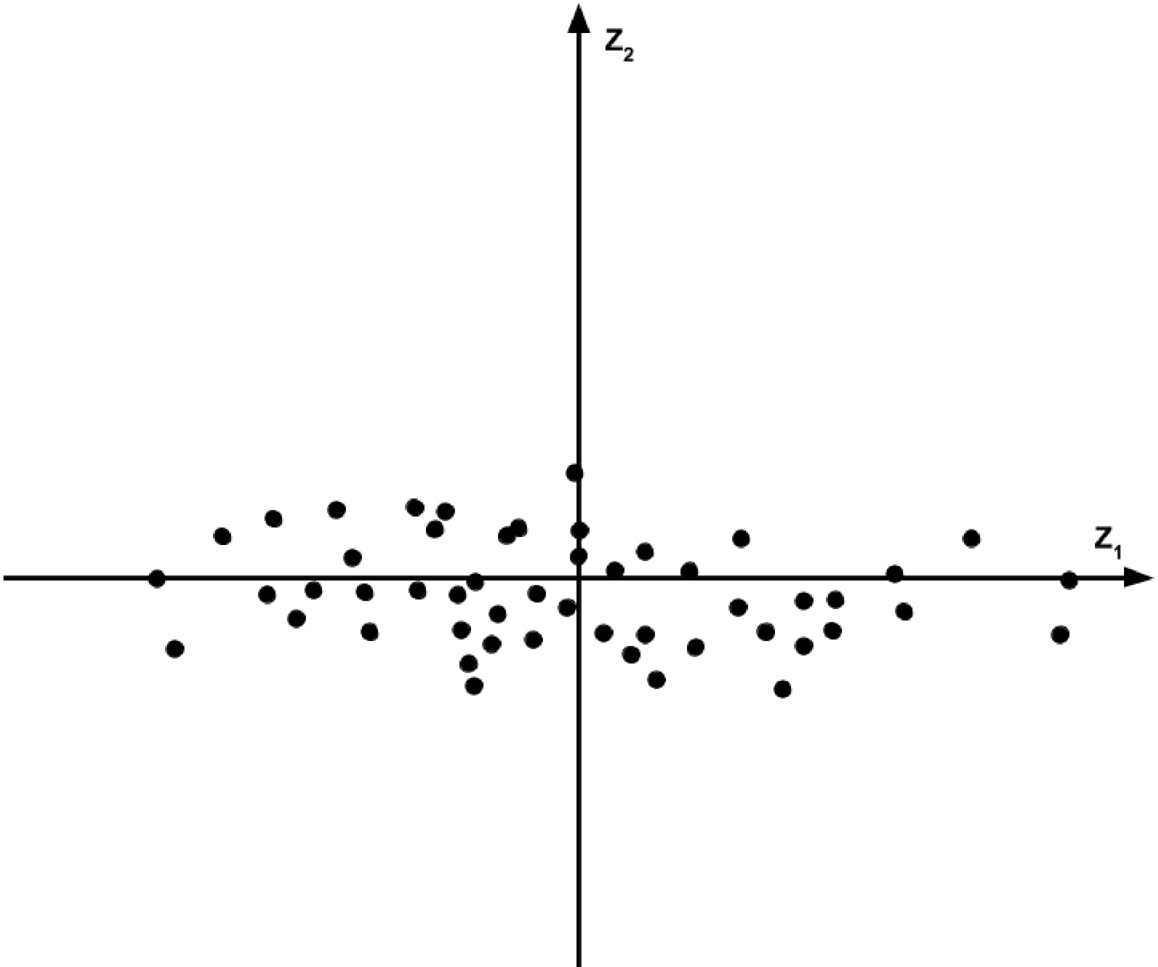


Figure 2.23: A two-dimensional data set projected into a space spanned by its two principal component vectors.

In Fig. 2.23, there is arguably just one principal component needed to “retain” most of the variance in the original data set. Thus, the second principal component vector may be discarded and the data set is reduced in dimensionality from two dimensions to one.

The data set, which can be defined as $\mathbf{Y}$, is mathematically defined as a matrix with n rows and m columns; each observation corresponds to a row and each column corresponds to the value of a feature for that observation. $\mathbf{Y}$ is z-scored to form $\mathbf{X}$, meaning that each column of $\mathbf{X}$ is scaled by subtracting from it its mean and then dividing it by its standard deviation. This ensures all data points are on the same scale.

The mathematical transformation of $\mathbf{X}$ is performed by constructing vectors

$$\mathbf{w}_{(k)} = (w_1, ..., w_m)_{(k)} \tag{2.42}$$

for k=1, ..., l where l is the maximum number of principal components chosen. Each component of $\mathbf{w_{(k)}}$ is simply a weight which determines how much of each of the original m basis vectors influences the resulting basis vectors. Each new basis vector is a linear combination of the original m basis vectors that is weighted such that the variance is maximized along the new basis vector while still being orthogonal all other principal component vectors. The first weight vector $\mathbf{w}_{(1)}$ is defined as

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w}\right)^2 \right\} \tag{2.43}$$

where $\mathbf{x}_{(i)}$ is a row of $\mathbf{X}$ and the dot product $\mathbf{x}_{(i)} \cdot \mathbf{w}$ calculates the so-called principal component score for each data point

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)} \qquad \text{for} \qquad i = 1, \ldots, n \qquad k = 1, \ldots, l \tag{2.44}$$

The summation is maximized when the spread of the data along $\mathbf{w}$ is maximal. Eqn. 2.43 can be rewritten in a form that involves the Rayleigh quotient.

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \{\|\mathbf{X}\mathbf{w}\|^2\} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{\mathbf{w}^T\mathbf{X}^{\mathbf{T}}\mathbf{X}\mathbf{w}\right\} \tag{2.45}$$

which is equivalent to

$$\mathbf{w}_{(1)} = \arg\ \max \left\{ \frac{\mathbf{w}^T \mathbf{X}^\mathbf{T} \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\} \tag{2.46}$$

since $\mathbf{w}$ is constrained to be a unit vector. $\mathbf{X}^\mathbf{T}\mathbf{X}$ happens to be a positive semi-definite matrix, and a standard solution is that the Rayleigh quotient is maximized when $\mathbf{w}$ is parallel with the eigenvector of the matrix. The maximal value for the Rayleigh quotient is the corresponding eigenvalue. The kth principal component is found by subtracting the first $k-1$ principal components from the data set given by $\mathbf{X}$ and then finding the unit vector which has the highest principal component score on the resulting data set,

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X} \mathbf{w}_{(s)} \mathbf{w}_{(s)}^{\mathrm{T}} \tag{2.47}$$

$$\mathbf{w}_{(k)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg\ \max \left\{ \tfrac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}. \tag{2.48}$$

Finally, the resulting transformed data set is given by

$$\mathbf{T} = \mathbf{X}\mathbf{W} \tag{2.49}$$

where $\mathbf{W}$ is a $p \times p$ matrix with rows being the weight vectors for each principal component in order of principal component number. Dimension reduction can then be performed by considering the eigenvalue corresponding to each principal component, and ignoring all but the first $m - L$ principal components to give

$$\mathbf{T}_L = \mathbf{X}\mathbf{W}_L \tag{2.50}$$

where $\mathbf{T}_L$ is an $n \times L$ matrix, which is reduced in the number of dimensions.

2.16 Self Organizing Map

A self organizing map (SOM) uses the idea of competitive learning that is inspired by processes that occur in complex biological systems such as the brain [37]. The topological organization of the brain is such that specific areas are stimulated by specific inputs more than others, forming a type of "map" with regions that could be associated with cognitive phenomena such as smell, memories, or joy. For example, the region of the brain associated with visual perception has maps for line orientation and color. This spatial organization of internal representations of information in the brain is what inspired a similar type of artificial learning system: the self organizing map.

2.16.1 Competitive Learning

Competitive learning first requires a set of statistical samples of a vectorial observable

$$\mathbf{x} = \mathbf{x}(\mathbf{t}) \in R^n \tag{2.51}$$

where $\mathbf{x}(\mathbf{t})$ indicates the vector at iteration t. In addition to the samples, or training data, there are the reference vectors given by

$$\mathbf{m_i}(\mathbf{t}) = \mathbf{m_i} \in R^n, i = 1, 2, ..., k. \tag{2.52}$$

Each reference vector is initialized randomly. Each of the k reference vectors are compared with the current training vector under consideration. Once the reference vector that is closest to the training vector is found, it is updated by some operation such that it becomes more aligned with the training vector. For example, the Euclidean distance is used as the criterion for determining the best matching reference vector. The result of repeating this process for a sufficient number of iterations is that eventually the reference vectors tend to point towards the points in space with the highest density of training vectors, and thus will approximate the underlying probability distribution. One such example of a method which produces an approximation to a continuous probability density function $\rho(x)$ of the vectorial input variable $\mathbf{x}$ is vector quantization. It achieves this by using a finite set of reference vectors and approximating $\mathbf{x}$ by finding the best matching unit (BMU). The BMU is the $\mathbf{m}_c$ where c is the index of the reference vector which minimizes

$$E = \int ||\mathbf{x} - \mathbf{m}_c||\rho(\mathbf{x})d\mathbf{x} \tag{2.53}$$

$$||\mathbf{x} - \mathbf{m}_c|| = \min_i ||\mathbf{x} - \mathbf{m}_i|| \tag{2.54}$$

Here $d\mathbf{x}$ is the volume differential in the $\mathbf{x}$ space. A closed-form solution for determining the optimal new direction for the BMU is not known and must be approximated using an iterative scheme. The alignment of $\mathbf{m}_c$ may be found using the method of steepest descent, which optimizes E and the result is

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)] \tag{2.55}$$

for some small monotonically decreasing $\alpha(t)$.

2.16.2 Self Organizing Map Algorithm

The algorithm is described below and the variables are defined as

- s - the current iteration
- λ - the iteration limit
- t - the index of the current input data vector from the training set
- $D(t)$ - the current input data vector
- v - the current weight vector of node v
- u - the index of the BMU for the current input data vector
- $\theta(u, v, s)$ - the neighborhood function, typically a Gaussian function of the current iteration s, which defines the distance from the current neuron within which all neurons are updated

- $\alpha(s)$ - the learning coefficient which monotonically decreases

The algorithm for the setup and training of the SOM is as follows:

1. Initialize the weight vectors for each node in the map to random weights
2. Randomly choose an input vector from the training set
3. For each node
   (a) Calculate Euclidean distance between the input vector and the node's weight vector
   (b) Keep track of the node which is closest to the input vector
4. Update all nodes within the neighborhood, defined by the function $\theta(u, s, v)$, according to Eqn. 2.55
5. Increment s by one and repeat from step 2 while $s < \lambda$

The weights, or components, of the SOM are modified according to

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s)) \tag{2.56}$$

The two quantities that affect which neurons are trained for a given BMU are $\theta(u, v, s)$ and $\alpha(s)$.

2.16.3 Training the SOM

A SOM is used to ultimately classify the texture of the MIP surface for this research since it allows for spatial segregation of responses into topologically related subsets. The training set is constructed by cutting each image of a MIP film into sections that are approximately 76 columns by 67 rows. A feature vector is then constructed based on the image and by using the features outlined in Sec. 2.12. The result of training a SOM is that it allows for the sorting of sections of images into a map where sections with similar texture and color are found close together. Using this process, defective sections can be detected.

2.16.4 Neuron Labelling

The neurons were given class labels by using functions provided by MATLAB such as "ginput()" to capture mouse clicking and "input()" to capture typed input. The function ginput() creates a crosshair on the image of the SOM shown in Fig. 2.24, and records the pixel coordinates of the current position when clicked. To convert the pixel coordinates to neuron indices, the pixel coordinates are divided by the corresponding cell dimensions; the x-coordinate is divided by the width of the image cells, and the y-coordinate is divided by the height of the image cells. This gives the number of cells to the right, and below, the origin. Once clicked, the script then prompts the user to put in the class label for the neuron. The script containing this process stores the class label in the corresponding element in the globally stored classifier matrix, and updates the image with a mark on the cell that was clicked to provide feedback to the user.

2.16.5 Classification of Image Sections

Using the "classifier" matrix outlined in the previous section, image sections are classified. Each image section is passed to the SOM, and the best matching unit, or the neuron that is most similar to the image section, is found. Once this neuron is found, the classifier matrix is used to determine the class of the image section by looking up the neuron's class label in the matrix. For example, if the neuron has class label 1, then the image section is classified as a defect. Additional class labels may be used to create a more complex classification system.

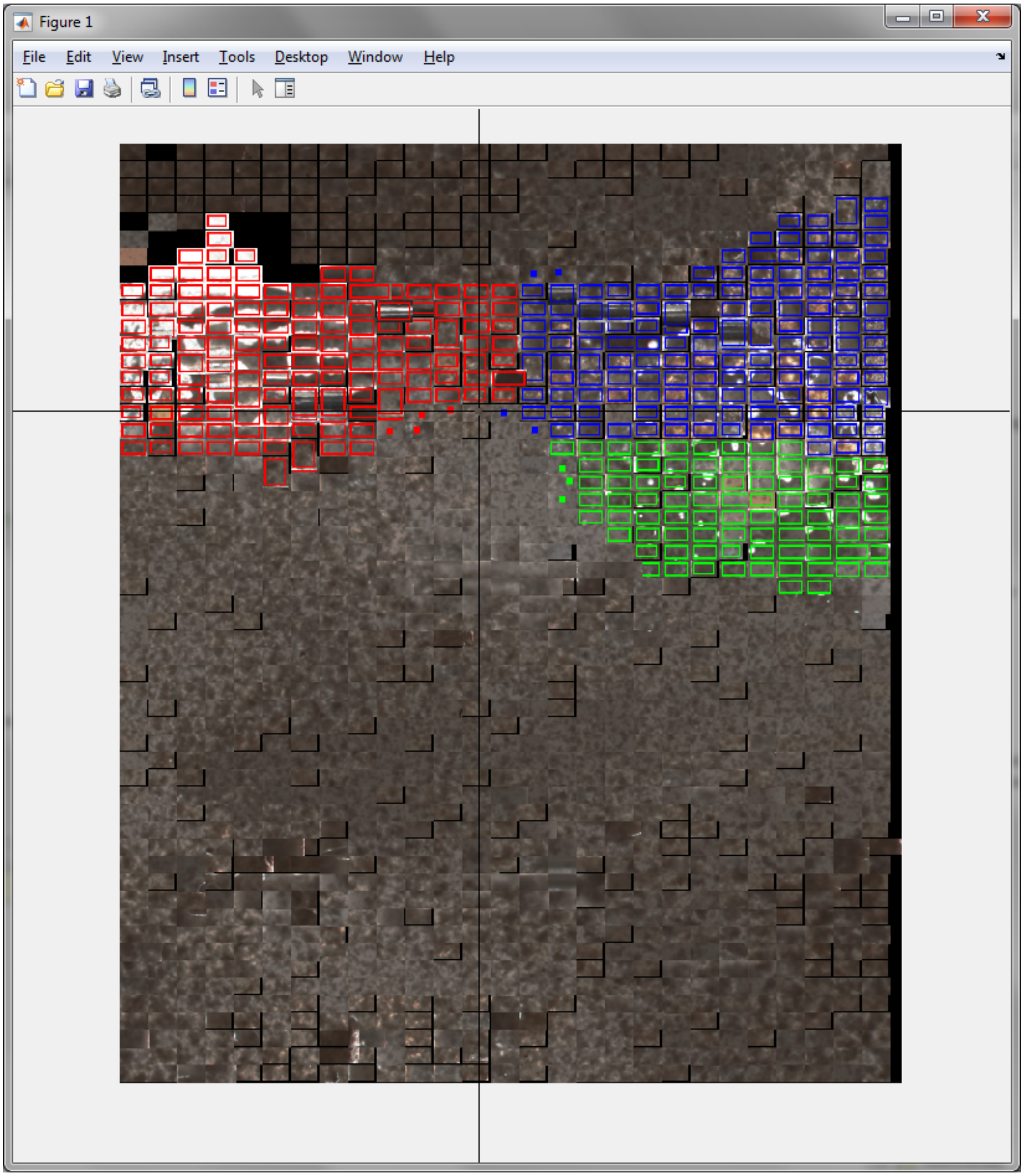


Figure 2.24: An example of a train SOM consisting of several hundred neurons. Cells for each neuron that have defect class labels are highlighted by color depending on which class they belong to.

Chapter 3

Results

3.1 A Processed Set of MIP Films

A total of 145 MIP films were imaged and processed using the described computational pipeline. For illustrative purposes, three of the MIP films which represent the data set were chosen and are presented in this chapter. The first MIP film was a good quality MIP with little to no scrapes or geometric defects, the second was a low quality MIP with several scrapes but no geometric defects, and the third had extreme defects in both texture and geometry. The films chosen to be processed are shown in Fig. 3.1, Fig. 3.2, and Fig. 3.3, which can be referred to as good, defective, and extremely defective respectively.

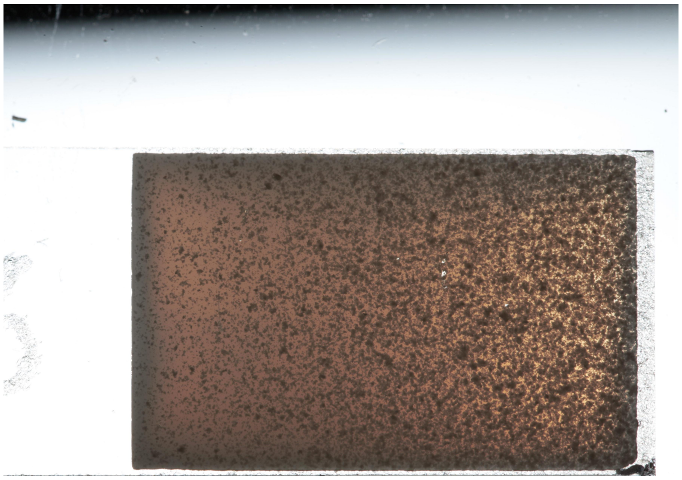


Figure 3.1: A picture of a good quality MIP film.

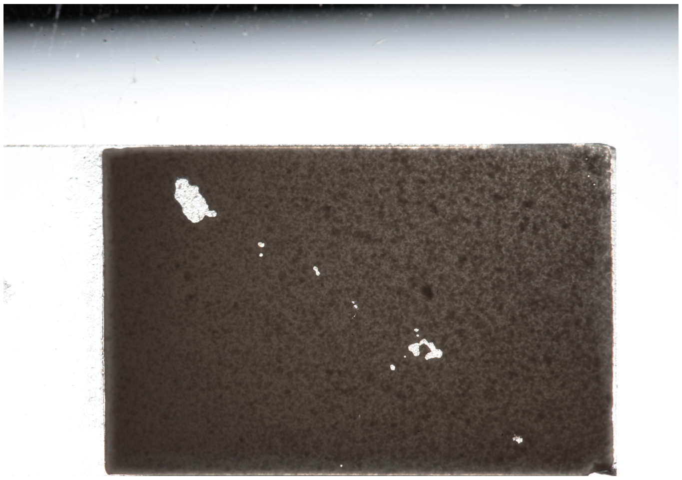


Figure 3.2: A picture of a defective MIP film.

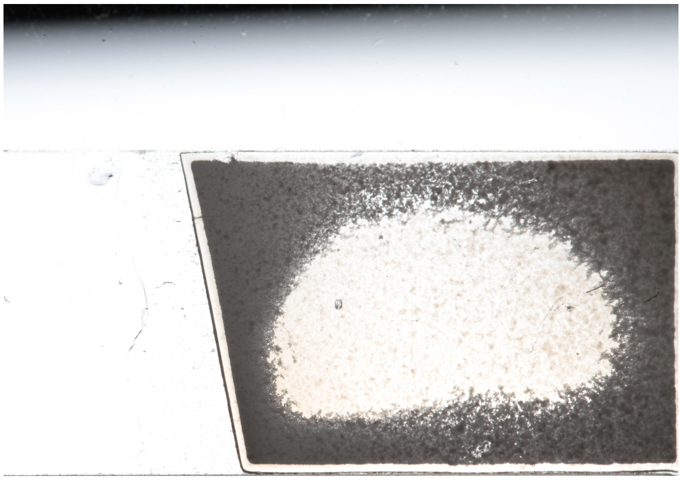


Figure 3.3: A picture of an extremely defective MIP film.

3.2 Detection of MIP Film Region

The next step in the computational pipeline was to determine the region of the image containing the MIP film. These regions were found using the process described in Sec. 2.5 and are visible as overlays in Fig. 3.4, Fig. 3.5, and Fig. 3.6.

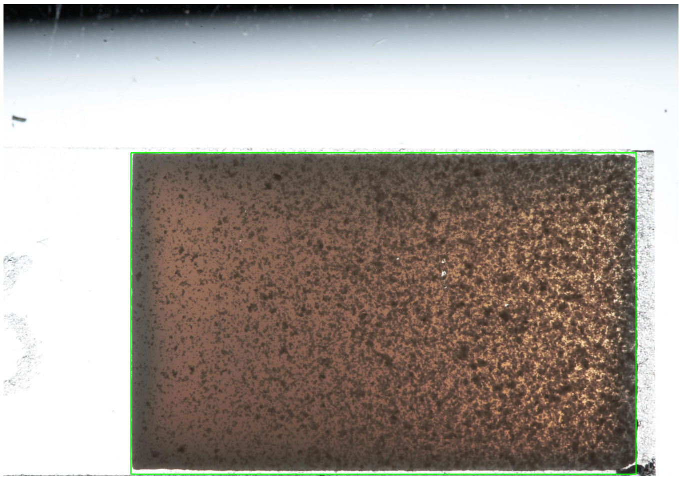


Figure 3.4: A good quality MIP film with its outline overlay.

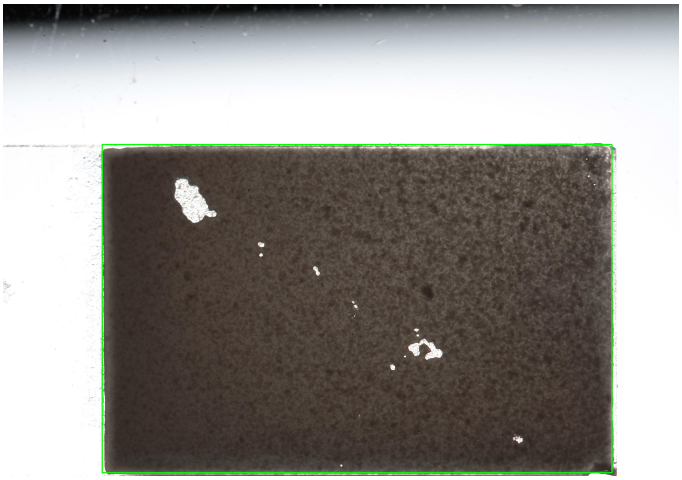


Figure 3.5: A defective MIP film with its outline overlay.

Edge trimming as described in Sec. 2.6 was applied to the MIP films prior to finding the final MIP-containing region. One example where it was applied with very noticeable results was the MIP film shown in Fig. 3.6. For this particular film, a consultation was had with a member from the Bottaro group based at MUN, the group who developed the tested MIPs, in which they believed the anomalous edge of the

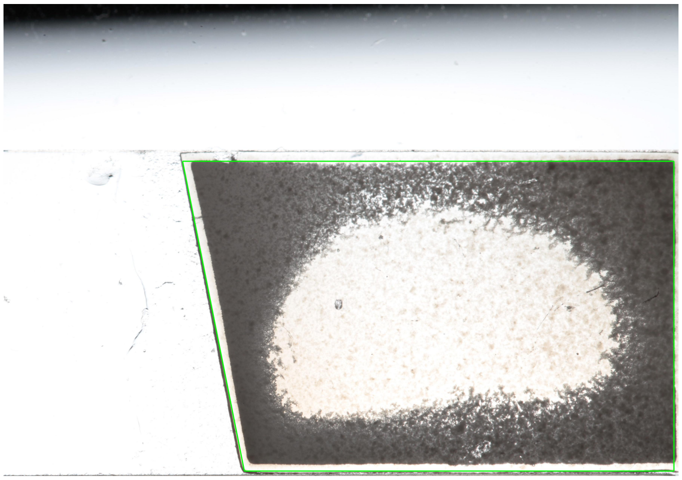


Figure 3.6: An extremely defective MIP film with its outline overlay.

MIP film was caused by residual solvent left over from the production process. Due to the odd shape of the MIP film and residue, edges from the residue and glass slide were chosen as two of the film sides. In this extreme scenario, the edge trimming algorithm refined the selected sides and obtained the final four lines outlined in Fig. 3.9. Figs. 3.7 and 3.8 had subtle differences from their respective untrimmed images. For example, the bottom line in Fig. 3.4 was correctly trimmed, but the difference with Fig. 3.7 was small.

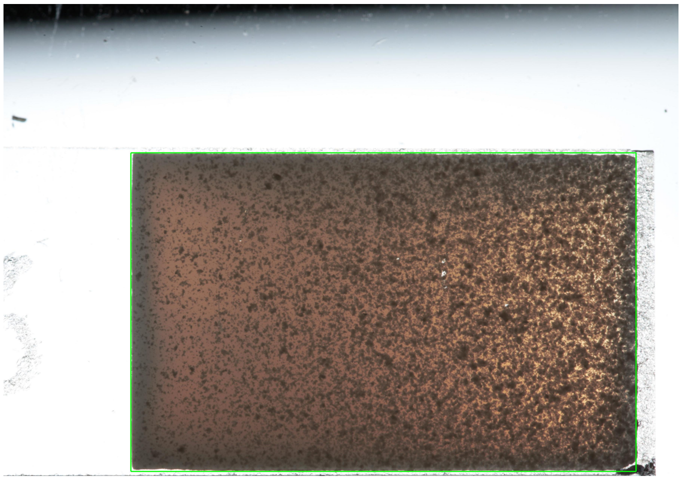


Figure 3.7: A trimmed good quality MIP film.

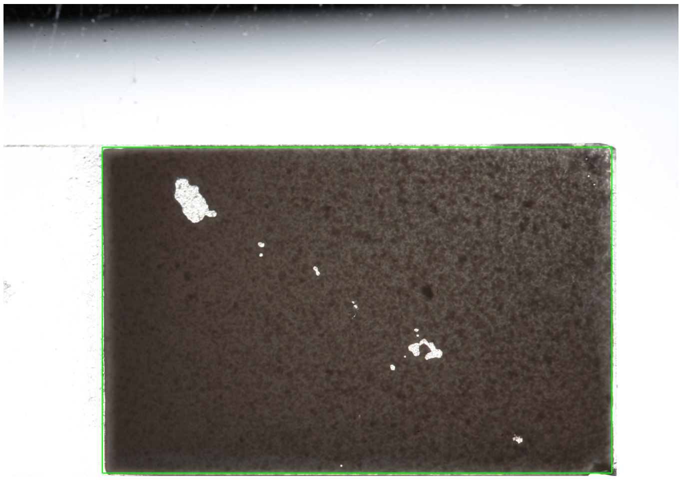


Figure 3.8: A trimmed defective MIP film.

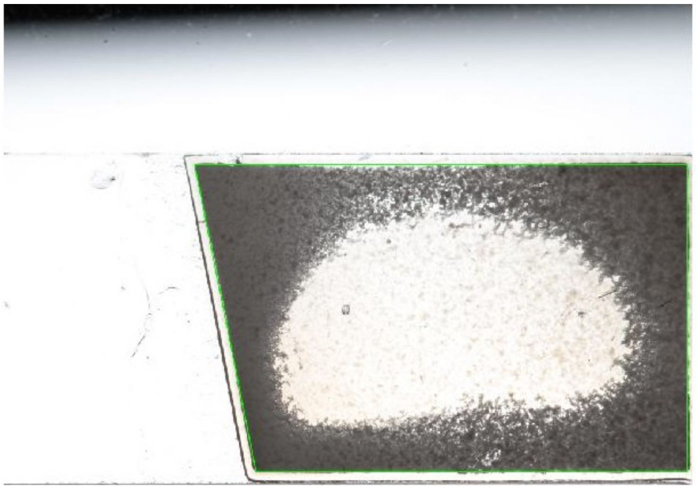


Figure 3.9: A trimmed, extremely defective MIP film.

3.3 Calculation of Geometric Properties

Geometric properties were calculated and overlaid on the three chosen MIP films as shown in Fig. 3.10, Fig. 3.11, and Fig. 3.12. The images, which had a size of 3744 rows by 5616 columns of pixels, had a scale of 153 pixels per millimeter.

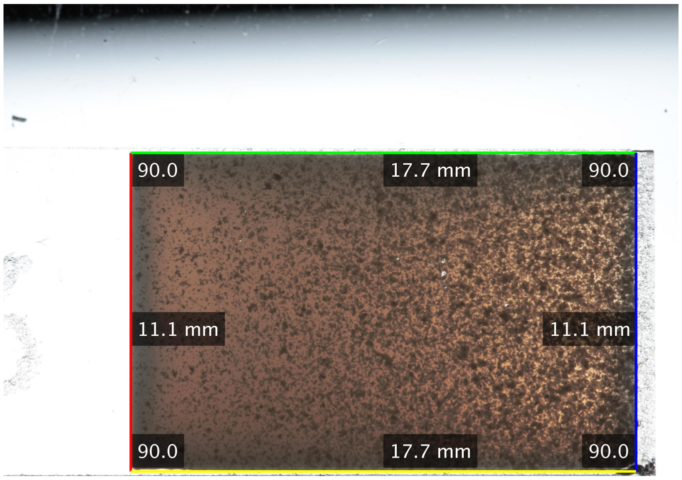


Figure 3.10: A good quality MIP film with its geometry overlay.

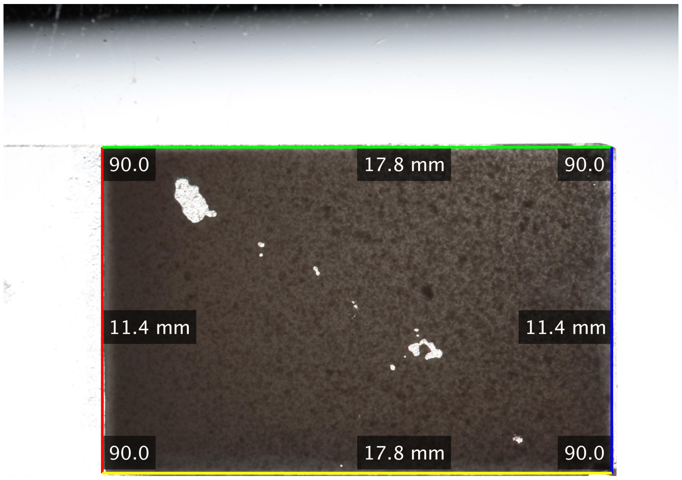


Figure 3.11: A defective MIP film with its geometry overlay.

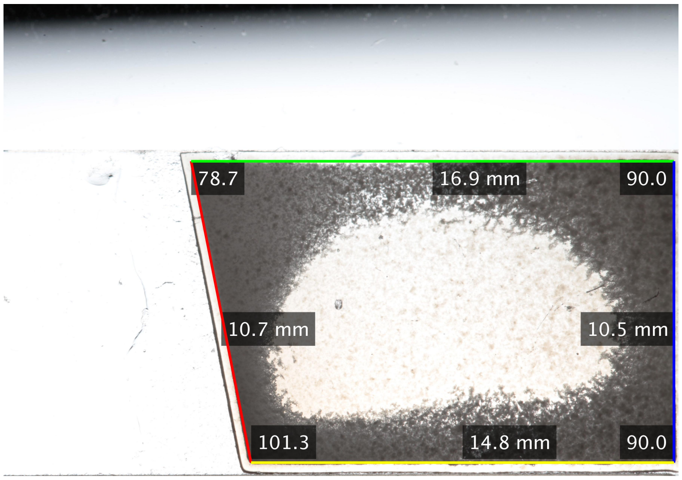


Figure 3.12: An extremely defective MIP film with its geometry overlay.

3.4 Measurement of Edge Quality

The strength, which is a measure of how sharp and clear an edge is, and regularity of the sides are of great importance since these characteristics affect the overall perception of quality of the MIP film. Using the Canny edge detector and Hough transformations, a measure of edge quality was obtained. As an example, a closeup of a MIP film was selected and is displayed in Fig. 3.13. By inspection, slightly weak edges appear to occur along the top edge of the film, while the edge appears stronger on the left side of the film. Furthermore, the top edge of the film also appears to be quite irregular. The first step in the algorithm is to divide each MIP side into line segments. For each segment, the regularity was calculated to obtain a "local regularity", that is, the regularity for a single segment along the side of the MIP as shown in Fig. 3.14.

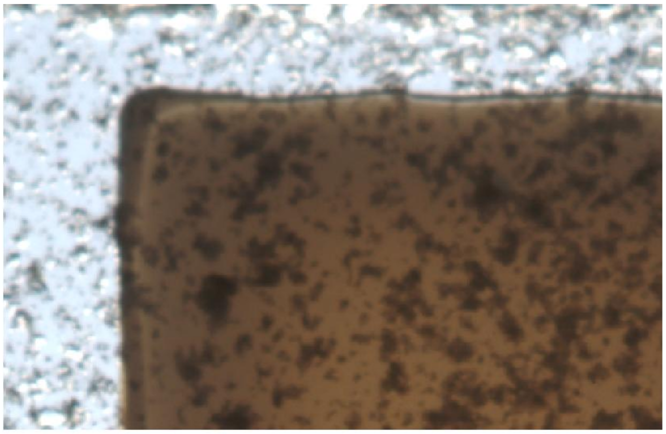


Figure 3.13: A close up of a MIP film corner.

In Fig. 3.14, each segment that was considered too irregular, meaning the average deviation of the edge pixels from the line was beyond some arbitrary threshold, was highlighted by a rectangle. Additionally, segments that contained few to no edge pixels were highlighted with cyan circles to indicate the presence of weak edge segments.

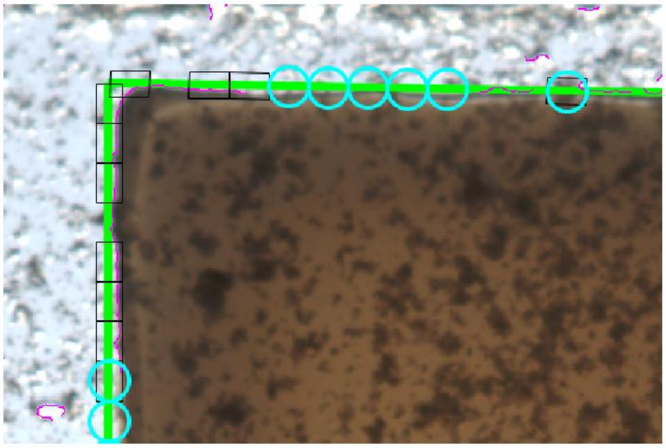


Figure 3.14: A close up of a MIP film corner with an overlay indicating the quality of its edge. Cyan circles indicate that the underlying line segment is weak, and black rectangles indicate that the underlying line segment is too irregular.

3.5 Feature Vector Creation

The SOM was trained on 50 images of MIP films only, each of which had a size of approximately 2600 × 1700 pixels, such as the film in Fig. 3.15. Each extracted film image was then cut into a 40×40 grid of cells. Cells had dimensions of approximately 76 pixel columns by 67 pixel rows. Two examples of such cells are shown in Fig. 3.16 and Fig. 3.17. The small scratch found just left of the lower-middle of the image in Fig. 3.15 is shown in Fig. 3.17. An example of a darker, good quality cell is shown in Fig. 3.16. Using a total of the 80,000 cells, a feature vector was calculated from each cell.

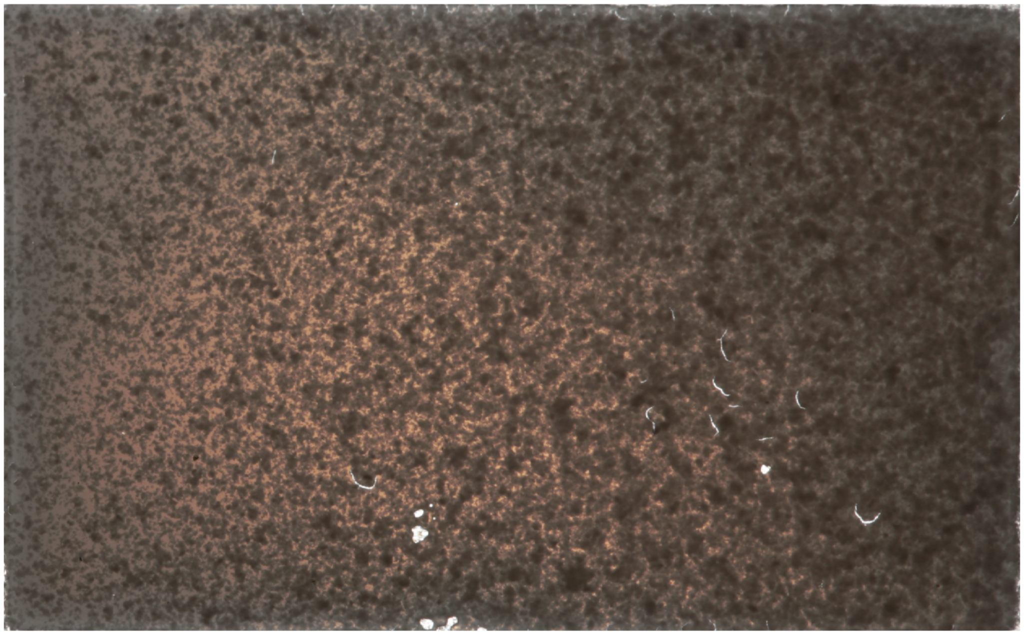


Figure 3.15: A typical MIP film used to train the SOM.

Figure 3.16: A closeup of typical film texture from the MIP film in Fig. 3.15

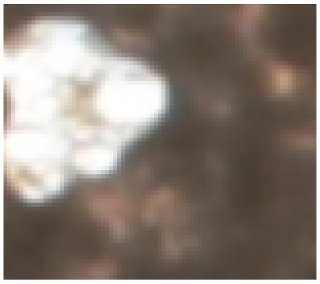


Figure 3.17: A closeup of a defect from the MIP film in Fig. 3.15

3.5.1 Principal Component Analysis

Principal component analysis was performed on a data set of 80,000 feature vectors, each vector having 20 components. Using the scree plot of the principal component eigenvalues with respect to principal component number shown in Fig. 3.18, the first four principal components were used for transforming the data set since the so-called elbow occurs between the third and fourth principal component in the scree plot.

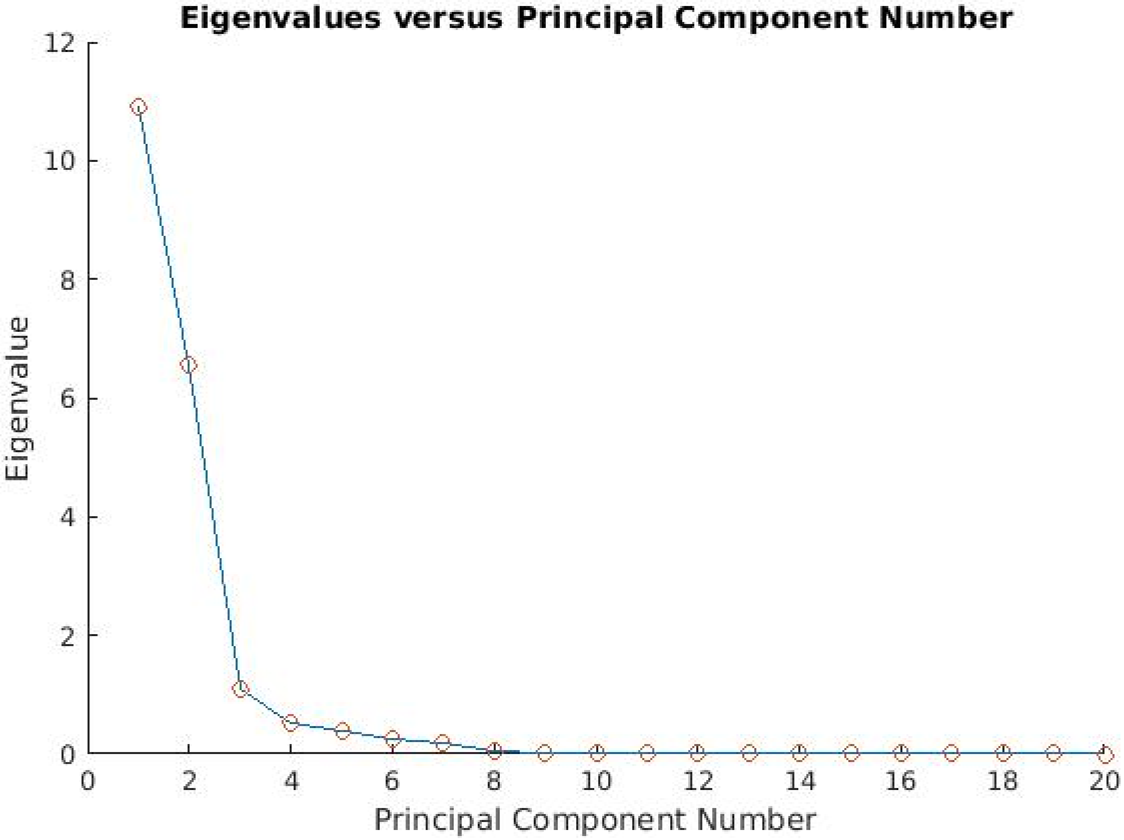


Figure 3.18: A scree plot of the principal component vectors obtained from applying PCA on the data set.

The components of the principal component unit vectors are listed below in Table 3.1.

Feature	PC1 Weight	PC2 Weight	PC3 Weight	PC4 Weight
Mean Gray Level	0.28	-0.10	-0.05	0.26
Median Gray Level	0.27	-0.12	-0.04	0.26
Energy	-0.18	-0.28	-0.14	0.33
Contrast	0.20	0.26	-0.28	-0.03
Variance	0.28	-0.15	-0.01	0.08
Homogeneity	-0.20	-0.26	0.27	0.04
Sum Average	0.28	-0.09	-0.05	0.31
Sum Variance	0.25	-0.21	0.01	0.11
Sum Entropy	0.21	0.27	0.04	-0.19
Entropy	0.21	0.27	0.03	-0.18
Difference Variance	0.20	0.26	-0.28	-0.03
Difference Entropy	0.20	0.28	-0.13	-0.01
IMC2	0.13	0.28	0.49	0.02
Maximal Correlation Coeff.	0.10	0.23	0.65	0.31
Blob0.5	0.26	-0.15	-0.06	-0.19
Blob0.6	0.24	-0.22	0.07	-0.26
Blob0.7	0.22	-0.24	0.12	-0.26
Blob0.8	0.21	-0.26	0.14	-0.24
Blob0.9	0.20	-0.27	0.14	-0.22
TTD	0.24	-0.01	-0.08	0.45

Table 3.1: A table of the weights of each principal component and their corresponding features.

3.5.2 Feature Vector Values for Various Textures

To provide further insights on the data set and what type of texture scores best with each of the chosen principal component vectors in Table 3.1, eight cells were chosen and are displayed in Fig. 3.19. For each image cell, its scores for each of the four principal components are listed in Table 3.2. Additionally, the values for the original feature vectors, before being transformed by PCA, are listed for each of the eight image cells in Tables 3.3 and 3.4. Finally, since the feature vectors are z-scored prior to being transformed by PCA, the mean and standard deviation of every feature for the data set of 80,000 image cells are listed in Table 3.5. The z-scoring of the feature values is a necessary step since some features have vastly higher ranges than others, such as mean gray level compared to homogeneity.

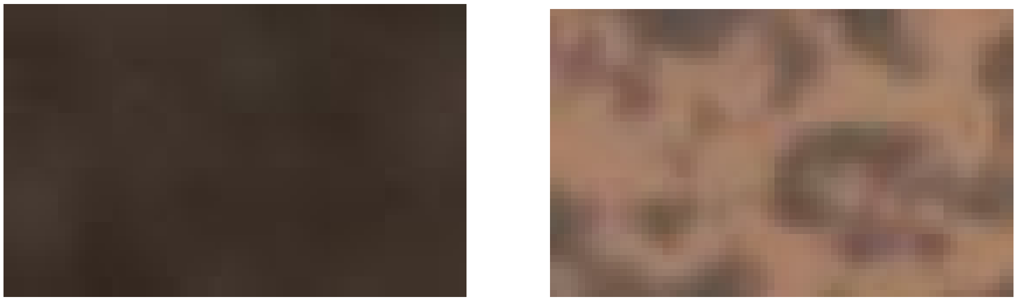
Some notable observations can be gleaned from this set of tables and figures. Firstly, very bright cells tend to score highly with the first principal component, such as cell H in Table 3.2, while dark cells such as cell A have a low score. Therefore, it seems that overall intensity is positively correlated with score on the first principal component.

Secondly, a high score with the second principal component seems to be correlated with, roughly speaking, how "speckled" the cell is, or how heterogeneously the pixel intensities are distributed. The most positively scoring cells for the second component are C and D which happen to be the most speckled cells in the set. The most negatively scoring cells for this principal component, A and H, are at the same time the most homogeneous cells in the set. Referring to Table 3.1, it is clear that some of the features the second principal component score is most positively correlated with are entropy and contrast. Entropy is high when pixel intensities are randomly distributed across the image, and contrast is high when local variations in pixel intensity are high.

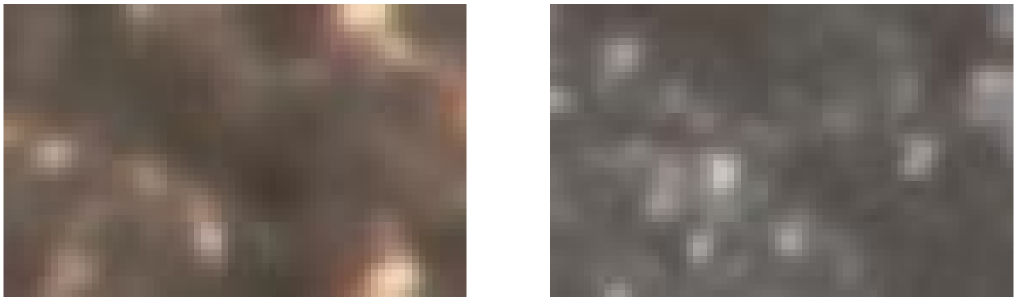
Finally, cells E and F are the highest scoring cells for the fourth principal component, which indicates that the fourth principal component favors cells containing neat and organized blobs of white pixels, which are most likely bubbles in the case of E and F. The fourth principal component's highest weight is on TTD, the custom feature which is high for images containing bright spots which change little with increasing intensity threshold. This implies that TTD is successful at distinguishing the presence of bubbles.

Cell	PC1 Score	PC2 Score	PC3 Score	PC4 Score
A	-4.80	-4.06	-2.79	-1.03
B	5.31	2.78	-2.37	0.27
C	3.91	3.55	-1.82	0.72
D	3.86	3.56	-2.07	0.58
E	0.77	0.50	-0.46	1.38
F	2.25	0.58	-0.59	2.08
G	5.73	2.92	-1.24	0.46
H	17.33	-18.16	2.76	0.25

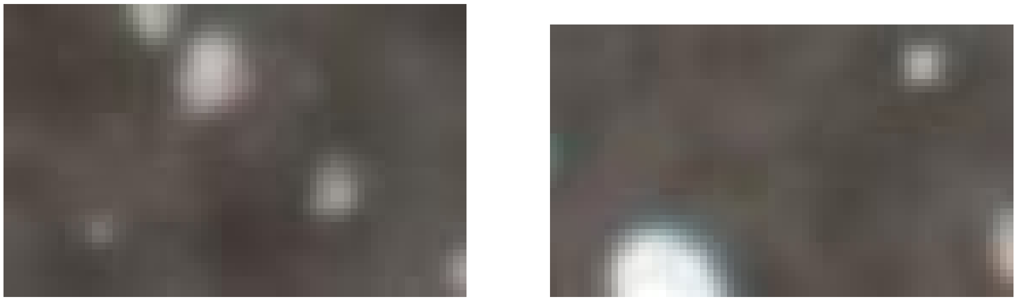
Table 3.2: The z-scored principal component scores for each of the MIP film cells shown in Fig. 3.19



(a) Cell A (b) Cell B



(c) Cell C (d) Cell D



(e) Cell E (f) Cell F

(g) Cell G (h) Cell H

Figure 3.19: Various MIP film image cells taken from a set of 80,000 image cells.

Feature	A	B	C	D
Mean Gray Level	51.42	119.04	98.04	99.37
Median Gray Level	51.00	122.00	94.00	96.00
Energy	1.00	0.34	0.37	0.35
Contrast	0	0.13	0.13	0.13
Variance	3.94	18.74	13.17	13.23
Homogeneity	1.00	0.93	0.94	0.94
Sum Average	4.00	8.60	7.13	7.19
Sum Variance	16.00	55.36	36.30	36.77
Sum Entropy	0	1.25	1.29	1.26
Entropy	0	1.94	1.99	1.94
Difference Variance	0.02	0.11	0.11	0.11
Difference Entropy	0	0.39	0.38	0.38
IMC2	0	0.87	0.90	0.88
Maximal Correlation Coeff.	0	0.83	0.90	0.85
Blob0.5	0	0.38	0.03	0.02
Blob0.6	0	0	0.02	7e-03
Blob0.7	0	0	8e-03	3e-03
Blob0.8	0	0	3e-03	0
Blob0.9	0	0	0	0
TTD	1.17	1.36	1.61	1.57

Table 3.3: The original feature vectors for MIP film cells A, B, C, and D shown in Fig. 3.19

Feature	E	F	G	H
Mean Gray Level	83.85	94.11	110.24	250.19
Median Gray Level	80.00	86.00	98.00	252.00
Energy	0.73	0.71	0.31	0.99
Contrast	0.06	0.07	0.12	1e-03
Variance	10.22	12.02	17.19	63.69
Homogeneity	0.97	0.96	0.94	1.00
Sum Average	6.32	6.66	7.93	15.99
Sum Variance	32.88	38.96	47.69	254.81
Sum Entropy	0.70	0.75	1.49	0.03
Entropy	1.07	1.16	2.28	0.04
Difference Variance	0.06	0.07	0.11	0.02
Difference Entropy	0.23	0.26	0.38	8e-03
IMC2	0.79	0.83	0.95	0.23
Maximal Correlation Coeff.	0.91	0.97	0.97	0.87
Blob0.5	0.03	0.06	0.15	1.00
Blob0.6	0.02	0.05	0.11	1.00
Blob0.7	5e-03	0.04	0.09	1.00
Blob0.8	0	0.03	0.07	1.00
Blob0.9	0	0.02	0.05	0.99
TTD	1.55	1.80	1.71	1.91

Table 3.4: The original feature vectors for MIP film cells E, F, G, and H shown in Fig. 3.19

Feature	Mean	Std
Mean Gray Level	79.24	26.61
Median Gray Level	77.82	26.62
Energy	0.61	0.21
Contrast	0.05	0.03
Variance	9.83	7.39
Homogeneity	0.98	0.02
Sum Average	5.97	1.66
Sum Variance	30.93	26.31
Sum Entropy	0.74	0.42
Entropy	1.11	0.63
Difference Variance	0.05	0.02
Difference Entropy	0.18	0.09
IMC2	0.75	0.24
Maximal Correlation Coeff.	0.83	0.22
Blob0.5	0.04	0.15
Blob0.6	0.03	0.12
Blob0.7	0.02	0.12
Blob0.8	0.02	0.11
Blob0.9	0.01	0.10
TTD	1.30	0.14

Table 3.5: The mean and standard deviation for each of the 20 features, calculated using all 80,000 cells.

3.5.3 Initialization and Training of the SOM

Using the suggested "trial-and-error" method by Kohonen [38] for determining the size of the SOM, the SOM was chosen to have 53 × 26 neurons. As a reminder to the reader, each neuron in a trained SOM possesses a particular weight vector, with neighbouring neurons tending to possess similar vectors. Fig. 3.20 shows a map of the SOM in which each neuron is represented by the cell from the training set that best matches the neuron's weight vector.

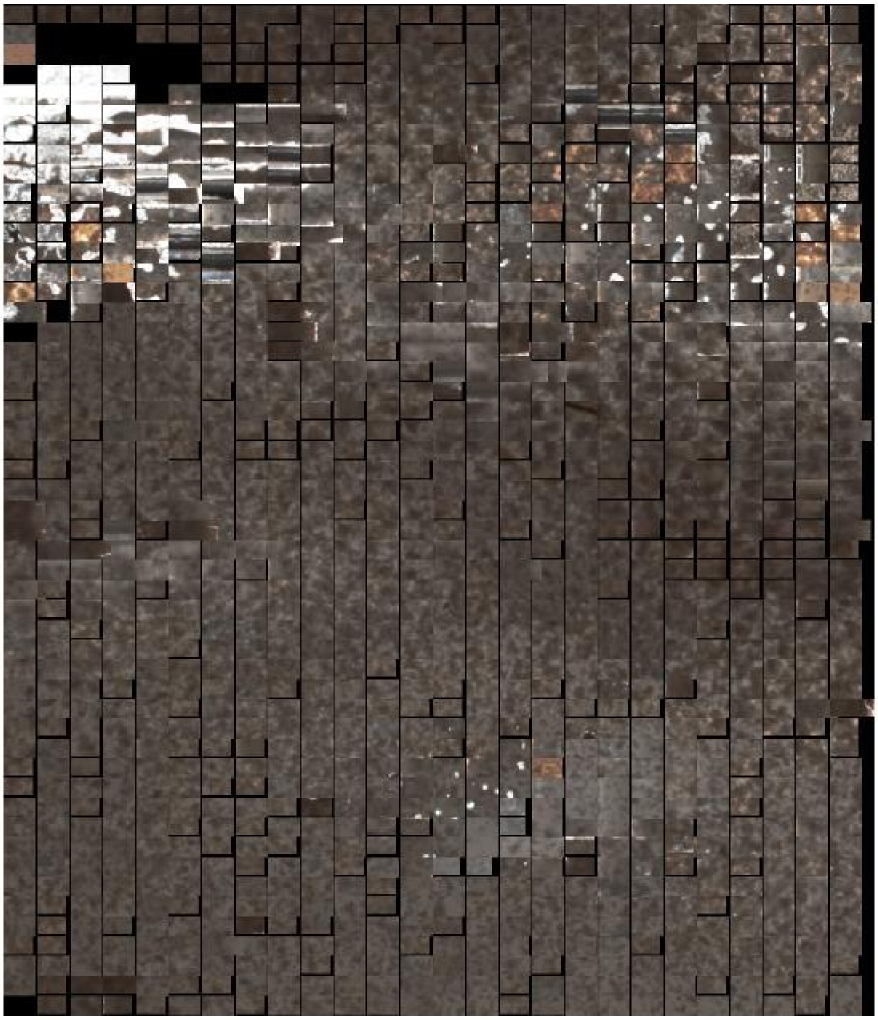


Figure 3.20: An image of the trained SOM with example cells for each neuron.

Cells containing scratches, or that were comprised solely of the glass slide, tended to congregate near the top left. Cells containing bubbles or "bubble-like" scratches tended to congregate near the top right. The lower two-thirds of the map tended to contain defect-free film sections with various degrees of speckleness. Additionally, the cluster in the top right appears to be made of two groups: the bottom being bubble-like scratches and the top being less defective but still unacceptable. Some anomalies do occur in the bottom-most portion of the map, where several slightly defective cells persist. The darkest cells tended to congregate at the top of the map. Lastly, certain neurons with no image cells, indicated by a black rectangle, indicated that not all neurons could be matched with an image cell after training was completed.

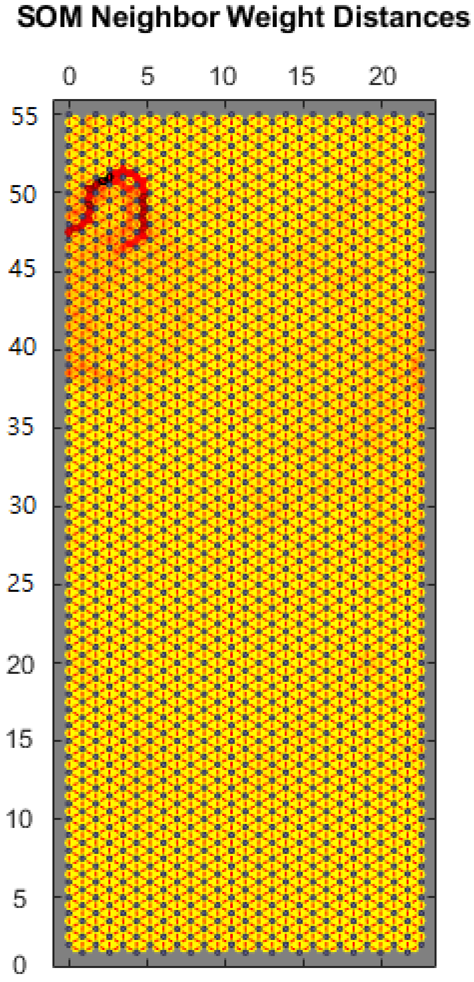


Figure 3.21: The neighbor weight distance plot for the trained SOM, with red indicating a large separation between neurons and yellow indicating minimal separation.

More insight regarding the distances between neurons may be derived by considering Fig. 3.21. Each dot in Fig. 3.21 corresponds to a neuron, and the color between each neuron indicates the euclidean distance between them. The most significant separation between neurons occurred near the top-left corner of Fig. 3.21. The dark-red curl in the top-left corresponds to the divide between the white cells and the dark cells in the top left corner of Fig. 3.20. There is also noticeable separation within the group of white, defective cells as well. Elsewhere, little separation can be seen between neurons according to Fig. 3.21.

3.5.4 Labelling Neurons

Neurons from the SOM that became associated with defective cells were given a specific label depending on which cluster they belonged to as shown in Fig. 3.22. The labelling of the neurons was done in a somewhat arbitrary fashion, with a neuron and its neighbors being labelled a particular class if they looked similar by eye. Certain clusters were obvious, and all neurons within it could be labelled the same, such as the cluster in the top left and the cluster in the bottom; they lay in a sea of fairly homogeneous gray and the division between the clusters and this sea were obvious. The cluster in the top right, however, seemed to be comprised of at least two different types of cells. Furthermore, this cluster could extend beyond what is currently labelled depending on the user's subjectivity.

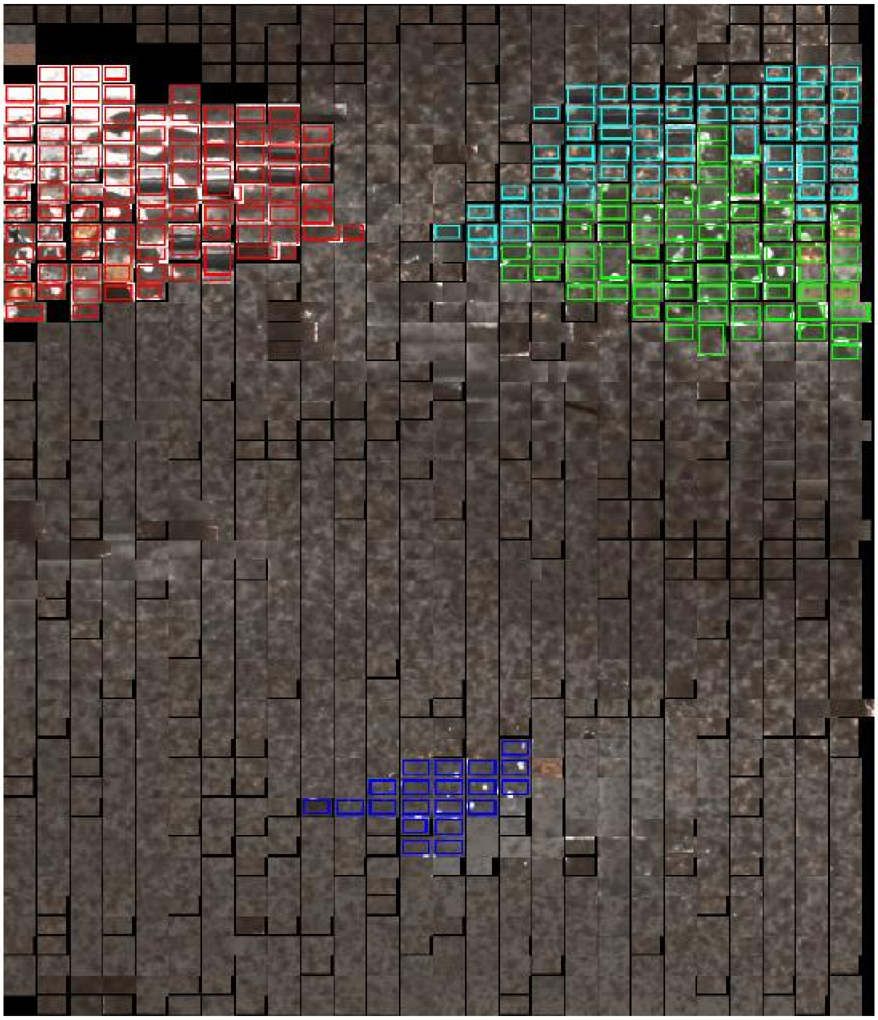


Figure 3.22: An image of the trained SOM with class labels for neurons associated with defects.

3.6 Texture Analysis

Finally, the textures of the three chosen MIP films were analysed by the pipeline and any cells containing texture associated with defects were flagged with a box that was colored corresponding to its class.

The good quality MIP film, shown in Fig. 3.23, had few obvious defects aside from an overly-speckled region in the right side. This was flagged by the SOM as being defective, evident by the abundance of cyan boxes. It appears, based on this example, that the cyan class, which may overlap with the green class, is most commonly comprised of thin cells which lie on the fringe between almost-acceptable and almost-defective. The cells that have been flagged with red boxes on the right-hand side of the film are likely the result of the image cell extending beyond the edge of the MIP film, indicated by the green line, which would result in the glass slide being included in the cell. This appears to be an artifact of how the MIP film was divided into cells, and it is correctable by setting the cell size to dynamically change to ensure only the interior of the MIP film is analysed. Given that the cell sizes are constant, and may not fit perfectly in the MIP film's interior, it is understandable why these cells were flagged as defective.

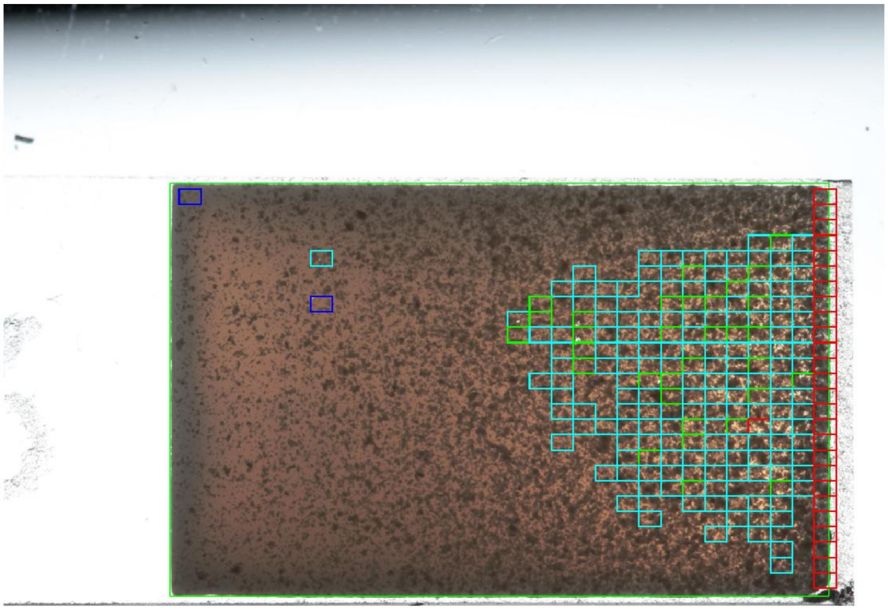


Figure 3.23: A good quality MIP film with flagged cells.

For the scraped MIP film, shown in Fig. 3.24, virtually all defective cells were flagged. The most obvious scratch in the top left corner was entirely flagged, and the remaining smaller scratches were almost entirely flagged. The cell in the lower right corner, labelled with a cyan box, appears to be a definitive scratch, which implies that the cyan class may also contain some scratches in addition to the "fringe" defects mentioned for Fig. 3.23. Additionally, similar to the good quality MIP film, cells on the edge of the film tended to be flagged as defects likely because either they extended beyond the Hough transform line, or the MIP film edge is irregular and does not perfectly coincide with the Hough transform line. In the latter case, this is visible in the cell in the top left corner where clearly the MIP film edge does not meet the Hough transform line. This indicates that careful analysis will need to take place for the cells that lie along the boundary of the MIP film, which will involve taking into consideration the irregularity of the edge within the cell.

Figure 3.24: A defective MIP film with flagged cells.

Finally, the extremely defective MIP film, shown in Fig. 3.25, has been labelled very well based on inspection. There does not seem to be any defective cell that is not labelled. Additionally, there is a pattern that white, heavily scraped cells were labelled red, slightly scraped and/or thin cells were green or cyan, and cells with small, almost imperceptible defects were blue. Again, the edge cells tended to be flagged as defective, likely due to the aforementioned reasons.

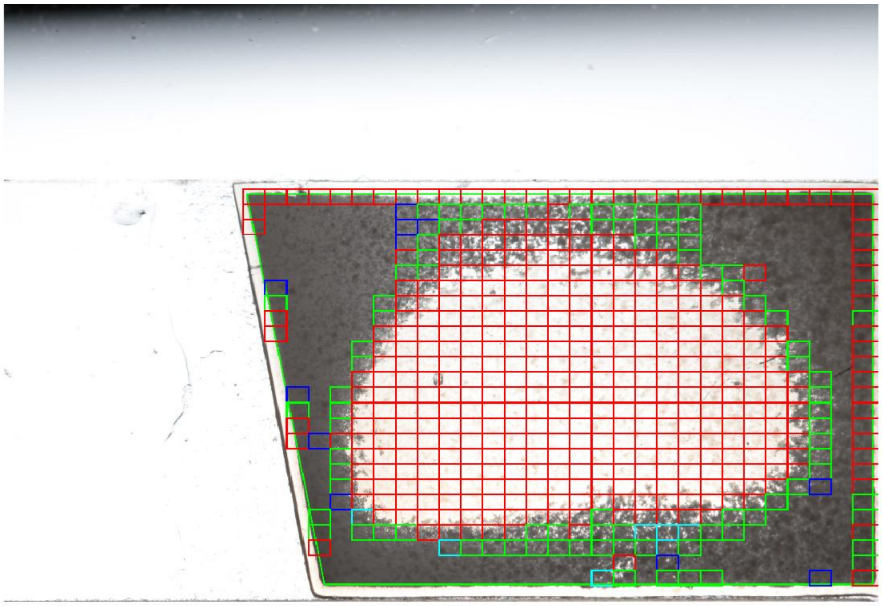


Figure 3.25: An extremely defective MIP film with flagged cells.

3.7 Scrape Detection

As a way of validating the performance of the SOM, a simple scrape detection algorithm was used to detect the presence of defects. After applying an intensity threshold to the three selected MIPs, the scrape detection algorithm was used to obtain results that were similar to the results obtained from the SOM. The algorithm is simple; given an image cell, it applies an empirically determined threshold of 0.75 to produce a black and white image. If the fraction of the total number of pixels in the cell which are white exceeds a threshold of 0.005 the cell is considered defective.

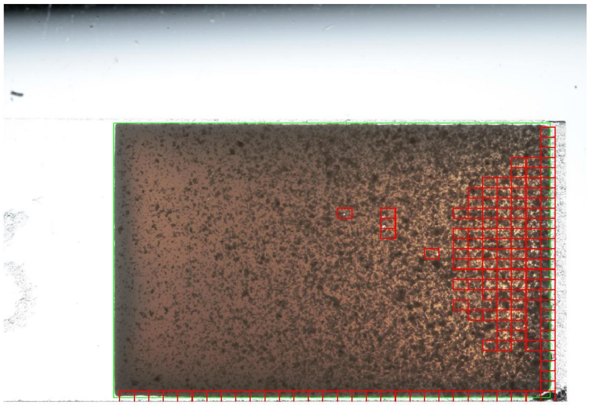


Figure 3.26: A good quality MIP film with cells flagged based on the scrape detection algorithm.

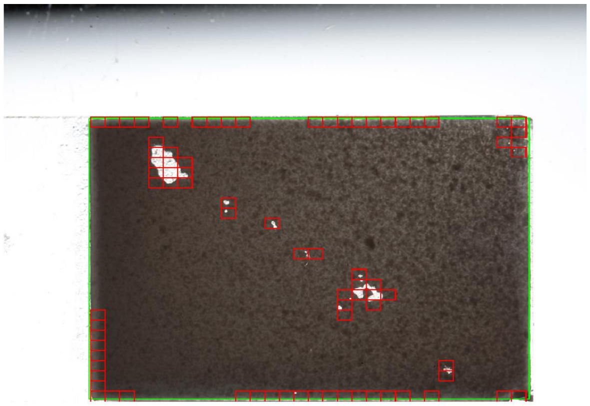


Figure 3.27: A defective MIP film with cells flagged based on the scrape detection algorithm.

The simple scrape detection algorithm and the more complex SOM approach were

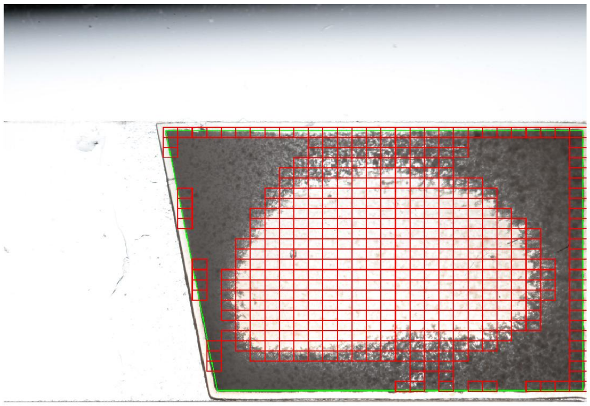


Figure 3.28: An extremely defective MIP film with cells flagged based on the scrape detection algorithm.

	SOM	Scrape
True Positive	0.09	0.07
True Negative	0.83	0.85
False Positive	0.03	0.01
False Negative	0.05	0.07

Table 3.6: A comparison of defect detection accuracy between simple scrape detection and the SOM-based approach. Each number in each column indicates the fraction of all MIP cells that were analysed, with the columns adding to one. True positive means a defect was correctly labelled, true negative means that a defect-free cell was correctly labelled, false positive means the cell was labelled defective when it was not, and false negative means the cell was defective but was labelled as not defective.

compared in terms of accuracies. A total of 301 cells were randomly chosen from the set of 80,000 image cells and labelled by hand. These same image cells were then classified by both approaches and cross-referenced with the human-assigned label. The results are shown in Table 3.6. In both approaches, the classification accuracy was 92%.

3.8 Statistics of a MIP Film Batch

Statistics were obtained for a batch of 145 MIP films which included geometric properties and prevalence of defects according to the SOM.

3.8.1 Statistics of Geometric Properties

Several geometric properties were obtained from the batch of 145 MIP films: surface area, aspect ratio, internal angles, edge weakness, and edge strength. Histograms of these properties are shown below.

The histogram shown in Fig. 3.29 appears to exhibit a bimodal distribution in surface area, although the number of samples in the 160-190 mm^2 range may not be enough to justify multimodality of the distribution. The largest peak occurs at around 200 mm^2 while the smaller peak occurs near 175 mm^2. It is possible that there are two populations of MIP films within the entire collection, which could be due to variations in creation of the glass slides, which influences the size of the film, or perhaps could be the result of different biases in film creation between researchers working in the lab.

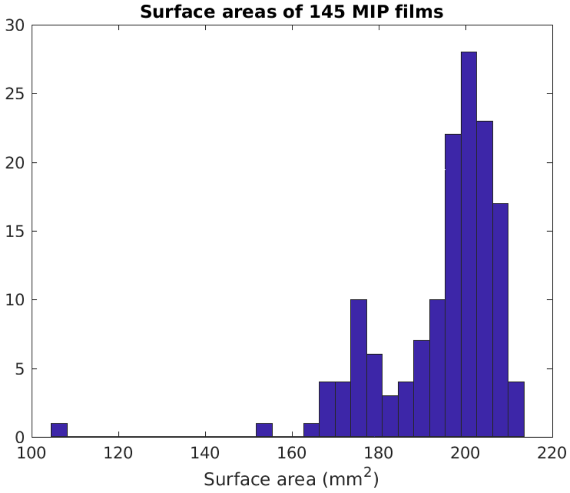


Figure 3.29: A histogram of surface areas of 145 MIP films.

The aspect ratios, which are plotted in Fig. 3.30, have a fairly uni-modal distribution aside from the odd peak for a specific range of values to the right of the main peak at approximately 1.66-1.68.

Edge weakness is plotted in Fig. 3.31, which indicates that a significant portion of the edges of all MIP films are fairly weak. For example, for MIP films that fell into the bin in which approximately 0.5 of the entire edge was considered too weak, this means that approximately 50% of all segments of the entire edge for each MIP film did not contain enough Canny edge pixels to be considered sufficiently strong. Edge

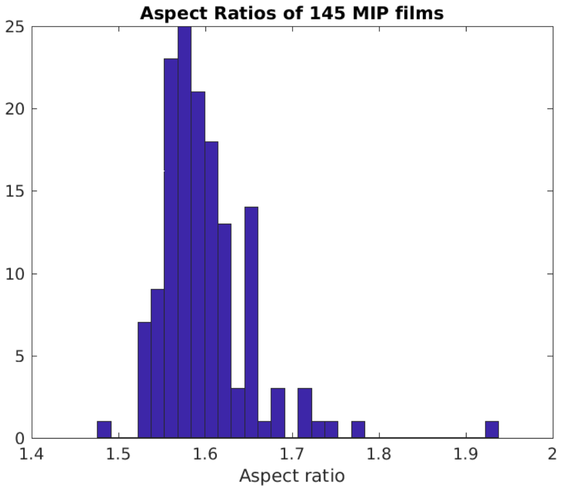


Figure 3.30: A histogram of aspect ratios of 145 MIP films.

irregularity, on the other hand, is quite high for a large population of MIP films and is plotted in Fig. 3.32. Approximately half of all MIP films had irregular edge segments for at least 50% of their edges. This means that a large portion of the MIP films' edges consisted of segments which deviated from the Hough transform lines beyond what was acceptable.

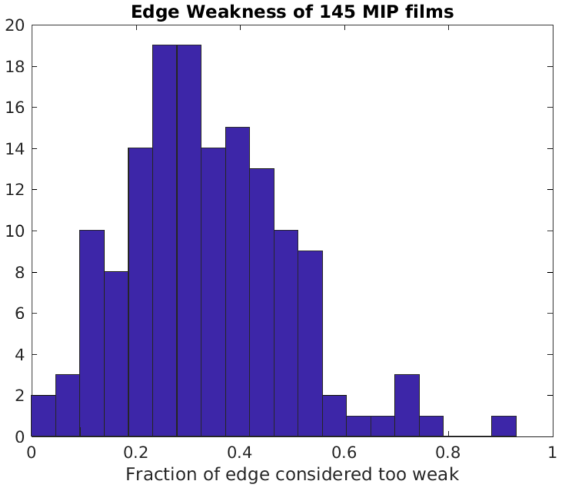


Figure 3.31: A histogram of edge weakness of 145 MIP films, which is a measure, between 0 and 1, of what fraction of the total edge is considered too weak.

Finally, internal angles for all MIPs are displayed in Fig. 3.33. Almost all angles fell very close to 90 degrees, with the exception of the two most extreme angles on the left and right sides of the histogram: 78.7 and 101.3, which were, coincidentally,

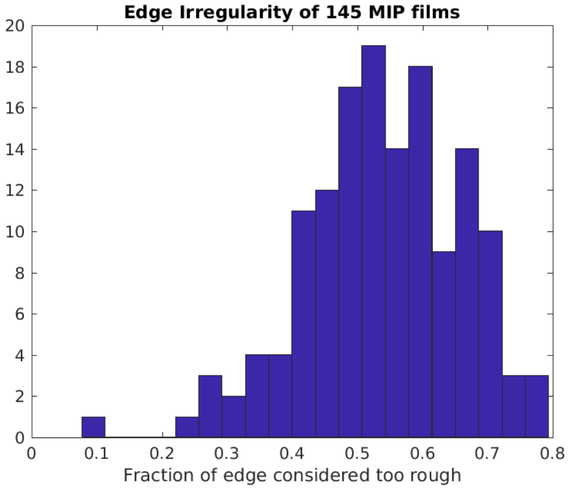


Figure 3.32: A histogram of edge irregularity of 145 MIP films, which is a measure, between 0 and 1, of what fraction of the total edge is considered too rough.

the two angles from the extremely defected MIP discussed in this chapter. These two angles are barely visible in the histogram as two small boxes.

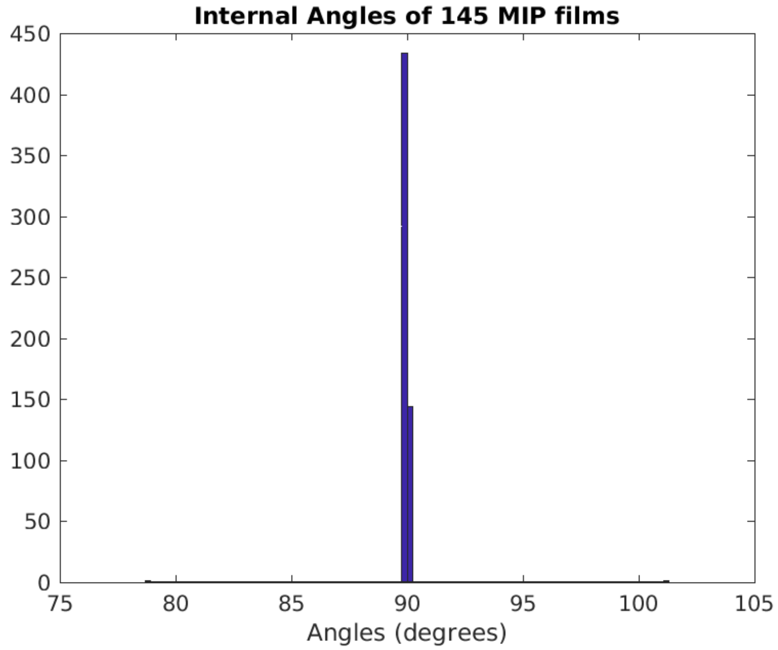


Figure 3.33: A histogram of internal angles of 145 MIP films.

3.9 Statistics on Defect Detection

In addition to analyzing the three chosen MIP films, all MIP films not used for training were analysed by both the SOM and simple scrape detection algorithm to obtain statistics on occurrence of texture defects and are plotted in a histogram in Fig. 3.34. Here, the fraction of defective film surface area is the fraction of the total area of a single MIP film which is defective. For example, in Fig. 3.34, 20 MIP films had between 2.5-5% of their total surface area labelled as defective.

The most noticeable difference between Fig. 3.34 and Fig. 3.35 is that the SOM defect rate is distributed more evenly throughout higher defect rate bins, and the scrape algorithm defect rate is heavily distributed in the low defect rate bins. This is consistent with Table 3.6 which indicates that, for the sample of several hundred image cells, the SOM tends to over-label cells as defective while the scrape detection algorithm tends to under-label cells as defective.

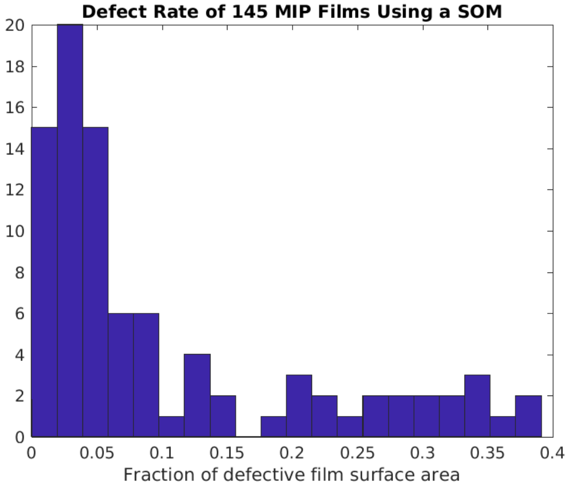


Figure 3.34: A histogram of defect rates, obtained from a SOM, in a batch of approximately 100 MIP films.

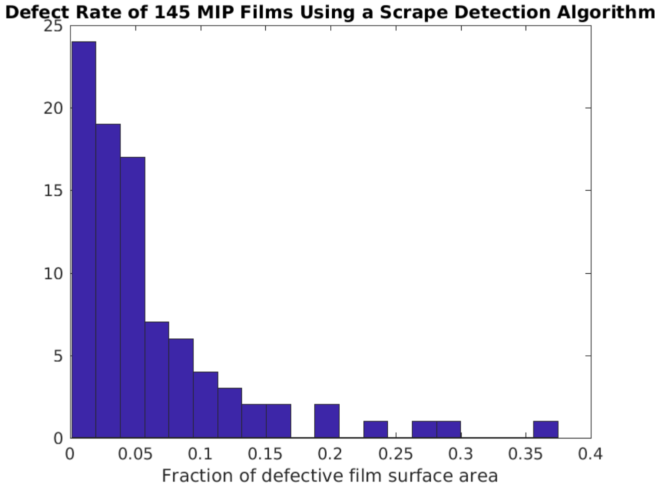


Figure 3.35: A histogram of defect rates, obtained from a scrape detection algorithm, in a batch of approximately 100 MIP films.

3.10 Effect of Changing Haralick Feature Offset

The Haralick texture features were calculated using a gray level co-occurrence matrix with an offset that compared the current pixel, I(i, j), with its neighboring pixel to the right as shown in Fig. 3.36. One potential issue with this is that the Haralick texture features will not capture vertical variation in texture and only horizontal variation in texture. Therefore, changing the offset to include multiple directions and distances was performed to see the effect on the trained SOM.

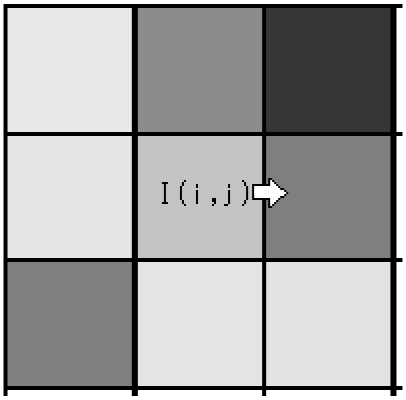


Figure 3.36: An illustration of the offset used in calculating the Haralick texture features for the original SOM.

A total of 16 offsets, indicated in Fig. 3.37 by the arrays of numbers, were used to create 16 individual gray level co-occurrence matrices. For every single image cell used to train the SOM, each Haralick texture feature was calculated using each of the matrices and the 16 resulting feature values were averaged to obtain a final Haralick texture feature that captured variation in texture in each of the 16 offsets.

The SOM was then re-trained using the same 80,000 image cells as before but with the new, averaged Haralick texture features. The result of the trained SOM is depicted in Fig. 3.38. The most noticeable effect of increasing the number of offsets is that the cluster of defected cells in the bottom-center of the SOM has been removed. However, a new, subtle cluster of slightly defective cells appears to have emerged in the lower-right corner of Fig. 3.38. Furthermore, the cluster in the top-right of the SOM appears to have been made significantly neater. There are three very distinct layers of cells: the top is filled with dark gray clumps atop a gray background, the middle consists of patchy cells with obvious bubbles, scratches, and other defects, and the bottom layer consists of more homogeneous cells with subtle gray clumps atop a

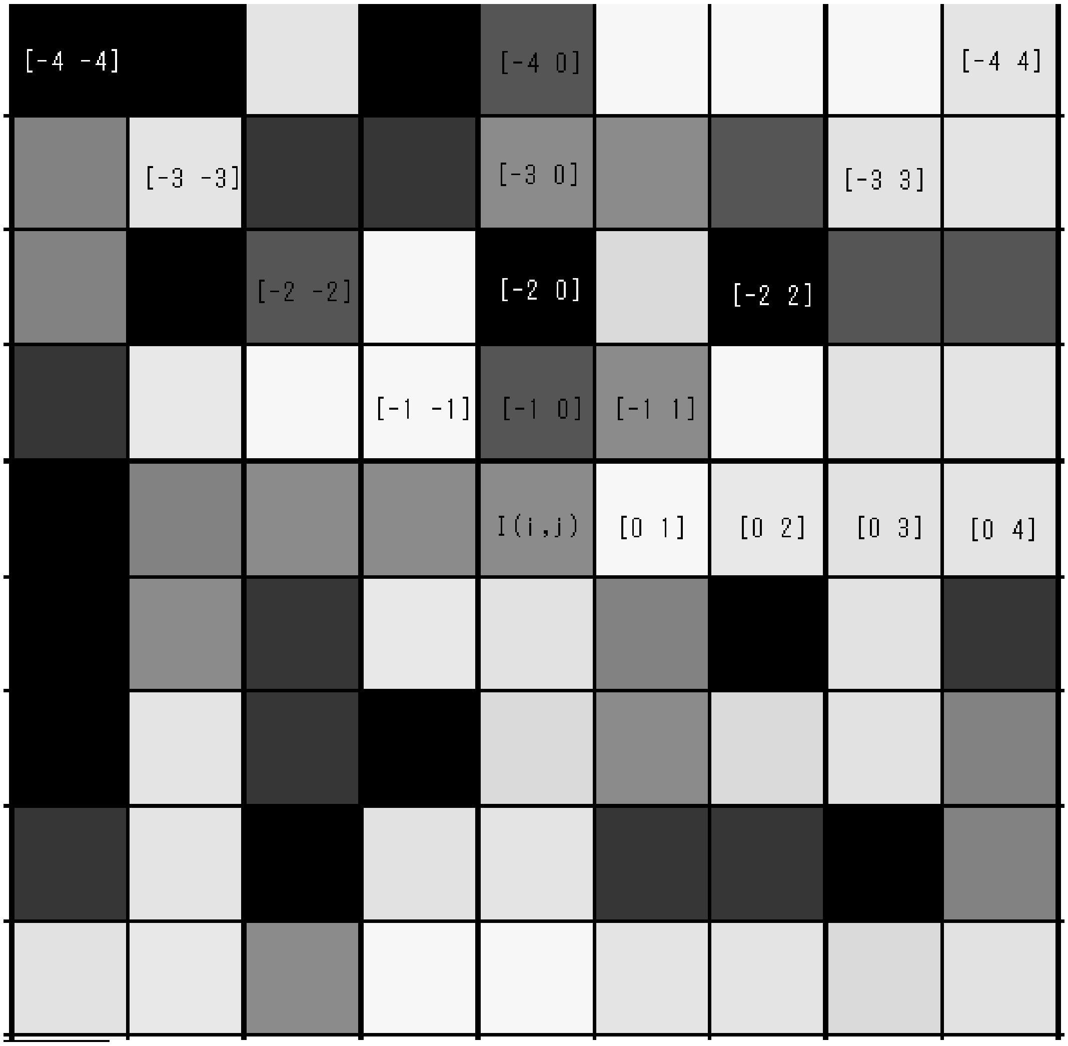


Figure 3.37: An illustration of the offsets used in calculating the Haralick texture features for a SOM.

solid light-gray background. The inclusion of multiple offsets reveals that more offsets leads to better separation of the data into more distinct clusters of cells. On the other hand, using multiple offsets increases computation time for calculating features, since for every offset a new GLCM must be computed. In this case, 16 GLCMs were calculated, and therefore 16 times the Haralick texture features were calculated for each cell. Whether the improved organization in the SOM is worth the additional computing time remains to be seen in future industrial application.

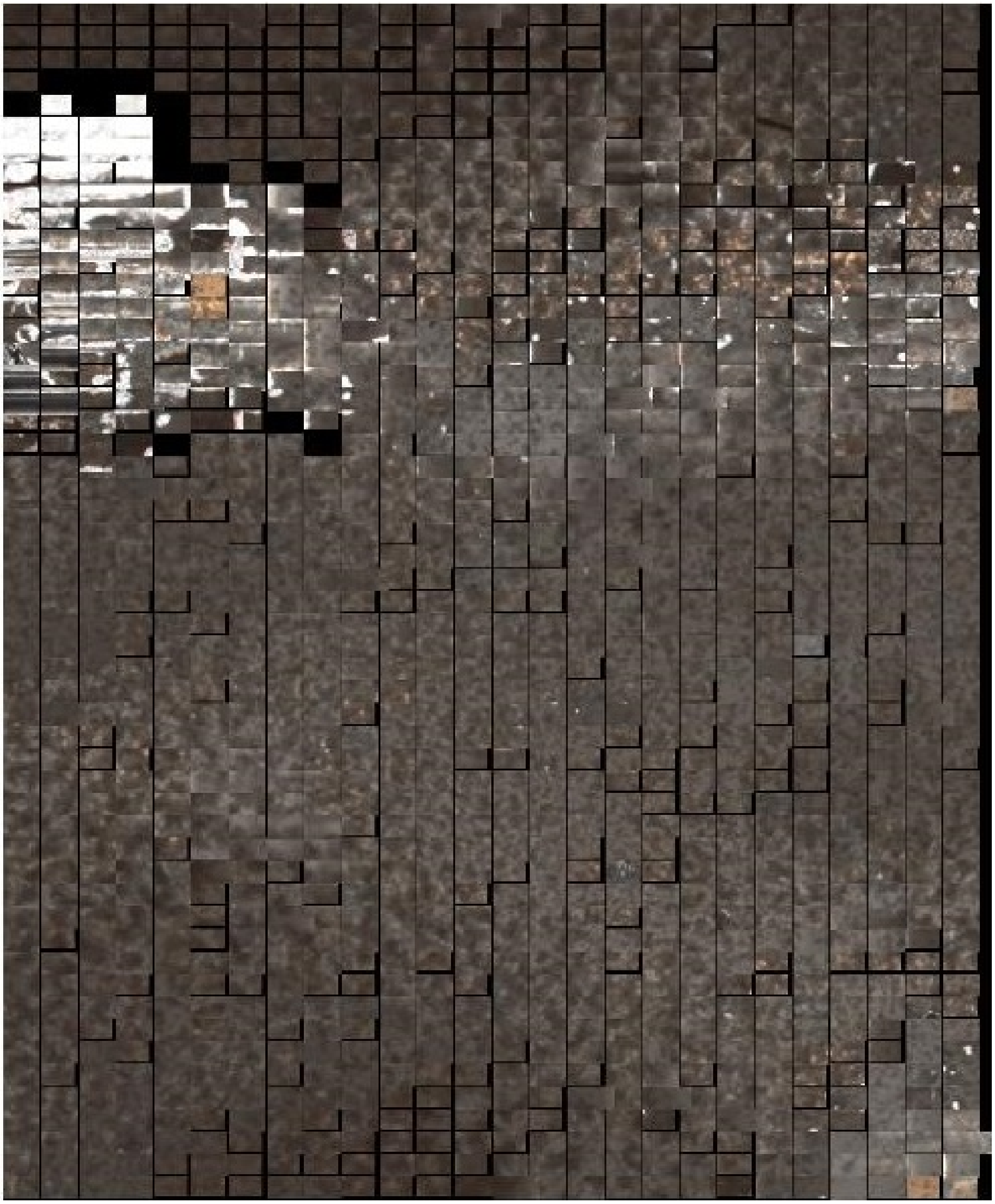


Figure 3.38: A representation of a SOM that was trained using multiple offsets for calculating Haralick texture features.

3.11 Effect of Increasing SOM Dimensions

Here we explore the effect of increasing the size of the SOM on the separation of clusters. The SOM used for texture analysis in this chapter had dimensions of 53 rows by 26 columns. A larger SOM was created from 60 rows by 32 columns of neurons and trained using the same training set as the original SOM. The result is shown in Fig. 3.39. Similar to changing the offset, increasing the size of the SOM caused the cluster in the bottom-center of the SOM to merge into another section of the map. Another cluster seems to have also appeared near the bottom-right corner of the MAP, and seems to consist of very small specks that appear in small numbers. The cluster in the top right also appears to have become more vertically dispersed, which indicates that increasing the SOM dimension along the vertical direction assists in increasing separation between classes of data. Finally, the cluster in the top left appears to have shrunk, or at least become more uniform, containing only cells that are entirely from the edge of the MIP film or are entirely white. Similar to increasing the number of offsets for computing Haralick texture features, greater separation of the clusters from increasing the SOM size comes at a cost of additional computing time. Likewise, whether this benefit is worth the additional computational complexity remains to be seen in industrial application.

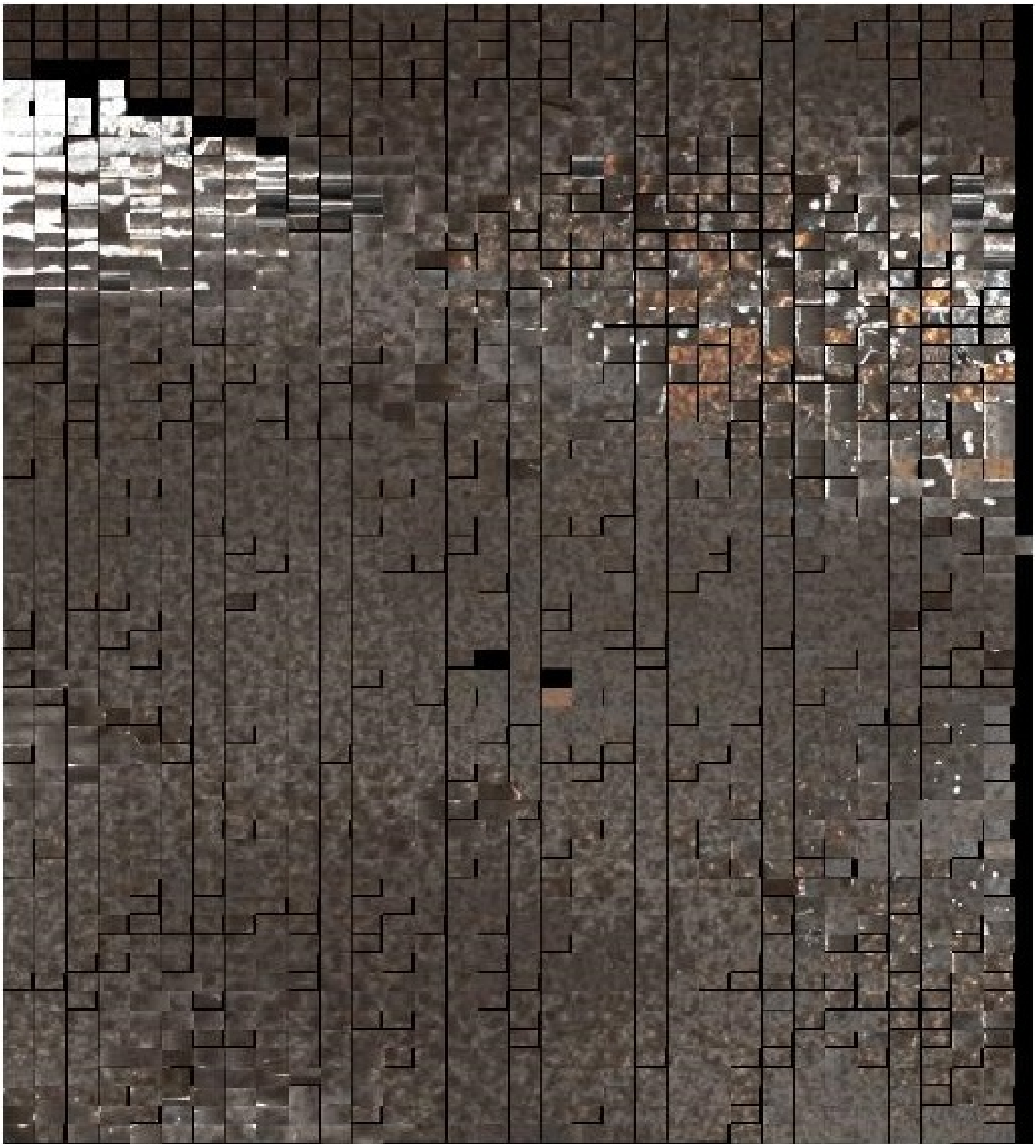


Figure 3.39: A representation of a trained SOM that contained more neurons than the original SOM.

Chapter 4

Discussion

The simple and intuitive approach to defect detection using the so-called scrape detection had similar defect detection accuracy as the approach using the SOM. A question that begs an answer is why one would ever use the SOM over the scrape detection approach. The first reason is that the SOM allows for more specific class labelling; an image cell containing a large scratch would be labelled a different class label than an image cell containing a large collection of small specks according to the SOM, whereas the scrape detection algorithm would just consider the total number of white pixels and label both cells as defective based on if the number surpasses a certain threshold. To elaborate, a large scratch composed of 100 white pixels will be "perceived" as the same type of image as an image with specks with total number of white pixels equal to 100, since the scratch algorithm only considers quantity of white pixels and not also the spatial distribution of those pixels. In other words, the scrape detection algorithm allows for a binary classification, the two classes being defective or non-defective, and the SOM allows for classification according to many classes and is able to find classes that may not be anticipated in a heuristic approach.

A second reason for using a machine learning algorithm like the SOM is not to just find defective cells, but also to find cells that are associated with high performance. Certain textures may be highly correlated with high performance, and so the SOM could potentially be used to label cells that are known to exhibit high performance rates. However, I would like to note that a regression-based machine learning algorithm may be better suited to predicting performance rate based on the texture of the image. This is due to the fact that regression-based algorithms find a continuous function which could map the texture of an image to a performance value which exists on a continuum, rather than having an algorithm, like a SOM, which returns discrete class labels for image cells. Alternatively, modern deep learning methods such as In short, the machine learning approach allows for a more complex labelling system which provides more options to the user of a quality control system such as the one described in this work.

In addition to exploring the use of regression-based machine learning algorithms, more modern techniques may also prove to be effective in detecting defects in the context of the quality control problem outlined in this thesis. For example, Fisher criterion-based stacked denoising autoencoders (FCSDA) have been used to find defects in fabrics for quality control in the textile factory, achieving a 95.83% classification accuracy for correctly identifying either defective or non-defective fabric samples [39]. Additionally, a deep convolutional neural network (DCNN) has recently been applied to the problem of detecting defects, such as so-called "bugholes", on concrete surfaces, which took into consideration the effects of illumination, shadows, and several other real-world imperfections on the dataset [40]. This method achieved a classification accuracy of 96.43% in classifying concrete surface cracks, bugholes, and color variation. Finally, a DCNN was used in detecting defects, such as chips, in so-called crossties in railway tracks with a classification accuracy of 92.11% [41]. Thus, advancements in the detection of defects in the context of this thesis' problem may be made by drawing inspiration and techniques from modern deep learning techniques.

The cluster of defective cells located in the upper right corner of the SOM illustration seems to have multiple types of cells that don't appear to belong together. This may be an issue of insufficient separation between the clusters caused by the choice of features used to describe the images. Another solution to alleviate this problem might be to increase the number of principal components for the final data set. A more rigorous exploration of the number of PCA components to use may also be a potential avenue of exploration, since the chosen number of retained components is done so through an art rather than a science. In future applications, the number of retained components can be chosen after the MIP manufacturers have provided input on the criteria they need characterized most accurately.

Another point to mention is the differences in the types of MIP films. Some

MIP films appear more red in shade than others, and some MIP films appear as a type of gray curd-like paste. The MIP films chosen were done so at random from a container of MIP films that have already been used by the Bottaro group. It is possible that the MIP films were composed of different polymer matrices, or were exposed to different types of solutions for testing that caused the discrepancies in the appearances. Finally, one may notice that in the histogram of surface areas the presence of a bi-modal distribution. This is likely caused by different individuals with their own biases in manufacturing the MIP films that caused the creation of more than one population of film.

Chapter 5

Conclusions

The computational pipeline described in this thesis has demonstrated capabilities of measuring geometric properties of MIP films, quantifying edge quality, and performing texture analyses using a SOM. The results obtained from this computational pipeline encourages further exploration and application in a real industrial setting. The classification accuracy of 92% is promising, and with further refining and tuning of parameters it is very likely this classification accuracy can be increased. Increasing the number of offsets for the Haralick texture features and the dimensions of the SOM seem to be suitable changes to make to increase separation between classes in the SOM. With the inclusion of marketing and chemistry teams, insights on the exact desirable set of geometric properties can be obtained and used to precisely screen and remove MIP films from being sold.

5.1 Future Work

Some future work that remains is to dynamically change the size of image cells according to change in texture. This means that whenever there is an abrupt change in texture, the image cell will be cut up into smaller cells and the texture for each subcell re-analysed. This type of recursive texture analysis can be used to increase the amount of information extracted for more interesting sections of the image and could potentially reduce computational time since sections of the image where the texture does not change much will have larger image cell sizes. Additionally, parameters that could be changed in future work include the offset used for calculating the Haralick texture features, dimensions of the SOM, and the number of principal components retained for training. In terms of new features, it is possible that the scrape detection algorithm's binary output could be used as a feature for the data set used to train the SOM.

An improvement on the current algorithm for texture analysis could be to account for cells on the edge of the film that are defective by resizing the cell if it is intersected by the line obtained by the Hough transform. The resizing would be performed so that the cell fits in the interior of the film, and does not overlap with the edge of the film. The edge quality properties would be used to assess the quality of that particular image cell instead. Furthermore, one additional measure for quantifying edge quality could be the "roughness" of the Canny edge for a given segment of the side. This roughness could be based on the standard deviation of the distances of the edge pixels from the line obtained from the Hough transform, and would give a better measurement of how rough a given line segment is, rather than a measure of how far the MIP edge is from the Hough transform line.

Finally, one interesting avenue of research could be to use a machine learning algorithm to predict performance of the MIP film. The performance of the MIP is based on how much and quickly it absorbs the target compound(s). Certain textures could be correlated with higher adsorption rates and these textures would therefore be highly desirable. Thus, a MIP film's performance could be evaluated without having to perform any extensive testing in a lab. I believe that a regression-based machine learning algorithm would be best suited for predicting performance based on the appearance of the image. The pipeline would be very similar with the exception that a regression algorithm would be used in place of a SOM.

Bibliography

[1] G. Z. Kyzas and D. N. Bikiaris. Molecular imprinting for high-added value metals: An overview of recent environmental applications. *Advances in Materials Science and Engineering*, 2014:1–9, 2014.

[2] P. Szatkowska-Wandas, M. Koba, P. Koliski, and M. Szablewski. Molecularly Imprinted Polymers Applications: A Short Review. *Mini-Reviews in Organic Chemistry*, 10:400–408, 2014.

[3] H. Zhang. Water-compatible molecularly imprinted polymers: Promising synthetic substitutes for biological receptors. *Polymer*, 55:699–714, 2014.

[4] D.-L. Huang, R.-Z. Wang, Y.-G. Liu, G.-M. Zeng, C. Lai, P. Xu, B.-A. Lu, J.-J. Xu, C. Wang, and C. Huang. Application of molecularly imprinted polymers in wastewater treatment: A review. *Environmental science and pollution research international*, 22:963–977, 2014.

[5] Z. Xu, S. Chen, W. Huang, G. Fang, H. Pingzhu, and S. Wang. Study on an on-line molecularly imprinted solid-phase extraction coupled to high-performance liquid chromatography for separation and determination of trace estrone in environment. *Analytical and bioanalytical chemistry*, 393(4):1273–1279, 2009.

[6] X. B. Luo, Y. C. Zhan, and T. X. M. al. Novel molecularly imprinted polymer using 1-(-methyl acrylate)-3-methylimidazolium bromide as functional monomer for simultaneous extraction and determination of water-soluble acid dyes in wastewater and soft drink by solid phase extraction and high performance liquid chromatography. *Journal of Chromatography A*, 1218(8):1115–1121, 2011.

[7] H. Sambe, K. Hoshina, and J. Haginaka. Molecularly imprinted polymers for triazine herbicides prepared by multi-step swelling and polymerization method: their application to the determination of methylthiotriazine herbicides in river water. *Journal of Chromatography A*, 1152(1):130–137, 2007.

[8] R. Say, A. Ersoz, H. Turk, and A. Denizli. Selective separation and preconcentration of cyanide by a column packed with cyanide-imprinted polymeric microbeads. *Separation and Purification Technology*, 40:9–14, 2004.

[9] Research Project: MIP-Based System for Water Quality Monitoring. `https://sensing.mun.ca/`, June 2017. Accessed: June 02 2019.

[10] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[11] P. Xie and J. Zhou. Rectangle Positioning Algorithm Simulation Based on Edge Detection and Hough Transform. *The Open Mechanical Engineering Journal*, 8:58–62, 2014.

[12] N. Zayed and H. A. Elnemr. Statistical Analysis of Haralick Texture Features to Discriminate Lung Abnormalities. *International Journal of Biomedical Imaging*, 2015:1–7, 2015.

[13] M. V. Suhas and B. P. Swathi. *Significance of Haralick Features in Bone Tumor Classification Using Support Vector Machine*, volume 478, pages 349–361. Springer Singapore, Singapore, 2019.

[14] V. Bhateja, A. Gautam, A. Tiwari, L. N. Bao, S. C. Satapathy, N. G. Nhu, and D.-N. Le. *Haralick Features-Based Classification of Mammograms Using SVM*, pages 787–795. Springer Singapore, Singapore, 2018.

[15] S. Minaee, A. Abdolrashidi, and Y. Wang. Iris recognition using scattering transform and textural features. In *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, pages 37–42, Aug 2015.

[16] H. Yong, Z. Chun-Xia, and W. Hong-Nan. Automatic Pavement Crack Detection Using Texture and Shape Descriptors. *IETE Technical Review*, 27(5):398 – 405, 2010.

[17] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski. Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 496–499, May 2008.

[18] T. Howley, M. Madden, M. O'Connell, and A. Ryder. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge-Based Systems*, 19:363–370, 2006.

[19] A. Rea and W. Rea. How Many Components should be Retained from a Multivariate Time Series PCA?, arXiv:1610.03588, 2016.

[20] R. B. Cattell. The Scree Test For The Number Of Factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

[21] O. Silvén, M. Niskanen, and H. Kauppinen. Wood inspection with non-supervised clustering. *Machine Vision and Applications*, 13(5):275–285, 2003.

[22] J.-L. Xu and D.-W. Sun. Identification of freezer burn on frozen salmon surface using hyperspectral imaging and computer vision combined with machine learning algorithm. *International Journal of Refrigeration*, 74:151–164, 2017.

[23] S. Marino, P. Beauseroy, and A. Smolarz. Weakly-supervised learning approach for potato defects segmentation. *Engineering Application of Artificial Intelligence*, 85:337–346, 2019.

[24] H. S. Naik, J. Zhang, A. Lofquist, T. Assefa, S. Sarkar, D. Ackerman, A. Singh, A. K. Singh, and B. Ganapathysubramanian. A real-time phenotyping framework using machine learning for plant stress severity rating in soybean. *Plant Methods*, 13:1–12, 2017.

[25] L. Jiang, B. Zhu, X. Rao, G. Berney, and Y. Tao. Discrimination of black walnut shell and pulp in hyperspectral fluorescence imagery using Gaussian kernel function approach. *Journal of Food Engineering*, 81:108–117, 2007.

[26] M. V. Boland and R. F. Murphy. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics (Oxford, England)*, 17(12):1213–1223, 2001.

[27] H. Joutsijoki, M. Haponen, J. Rasku, K. Aalto-Setl, and M. Juhola. Machine Learning Approach to Automated Quality Identification of Human Induced Pluripotent Stem Cell Colony Images. *Computational and Mathematical Methods in Medicine*, 2016:1–15, 2016.

[28] M. A. Olsen, E. Tabassi, A. Makarov, and C. Busch. Self-Organizing Maps for Fingerprint Image Quality Assessment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 138–145, June 2013.

[29] A. B. Snehamoy Chatterjee, B. Samanta, and S. K. Pal. Image-based quality monitoring system of limestone ore grades. *Computers in Industry*, 61:391–408, 2010.

[30] D. S. Bulgarevich, S. Tsukamoto, T. Kasuya, M. Demura, and M. Watanabe. Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures. *Scientific Reports*, 8:2078, 2018.

[31] F. Anabitarte, J. Mirapeix, O. M. C. Portilla, J. M. Lopez-Higuera, and A. Cobo. Sensor for the Detection of Protective Coating Traces on Boron Steel With AluminiumSilicon Covering by Means of Laser-Induced Breakdown Spectroscopy and Support Vector Machines. *IEEE Sensors Journal*, 12:64–70, 2012.

[32] R. Szeliski. *Computer Vision: Algorithms and Applications.* Springer-Verlag, Berlin, 2010.

[33] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[34] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[35] E. Miyamoto and T. Merryman. *Fast Calculation of Haralick Texture Features.* Human Computer Interaction Institute, 2005.

[36] I. T. Jolliffe. *Principal Component Analysis.* Springer, New York, 2002.

[37] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[38] T. Kohenen. *MATLAB Implementations and Applications of the Self-Organizing Map.* Unigrafia Oy, Helsinki, Finland, 2014.

[39] Y. Li, W. Zhao, and J. Pan. Deformable Patterned Fabric Defect Detection With Fisher Criterion-Based Deep Learning. *IEEE Transactions on Automation Science and Engineering*, 14:1–9, 2016.

[40] G. Yao, F. Wei, Y. Yang, and Y. Sun. Deep-Learning-Based Bughole Detection for Concrete Surface Image. *Advances in Civil Engineering*, 2019:1–12, 2019.

[41] X. Giben, V. M. Patel, and R. Chellappa. Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 621–625, 2015.

Appendix A

Matlab Code

An up-to-date copy of the code used for this work can be found at
https://github.com/AndrewWay/mipqc.git

Appendix A

Image EXIF Meta Data

Image Width	5616
Image Height	3744
Bits Per Sample	8 8 8
Compression	JPEG (old-style)
Make	Canon
Camera Model Name	Canon EOS 5D Mark II
Preview Image Start	53696
Orientation	Horizontal (normal)
Preview Image Length	1208196
X Resolution	72
Y Resolution	72
Resolution Unit	inches
Modify Date	2019:03:22 10:54:46
Artist	Photographer: Marek Bromberek
Copyright	Copyright: Marek Bromberek
Exposure Time	1/30
F Number	9.0
Exposure Program	Manual
ISO	100
Exif Version	0221
Components Configuration	Y, Cb, Cr, -
Shutter Speed Value	1/32
Aperture Value	9.1
Exposure Compensation	0
Metering Mode	Multi-segment
Flash	Off, Did not fire
Focal Length	100.0 mm
User Comment	
Sub Sec Time	18
Sub Sec Time Original	18
Sub Sec Time Digitized	18
Flashpix Version	0100
Color Space	Uncalibrated
Exif Image Width	5616
Exif Image Height	3744
Focal Plane X Resolution	3849.211789
Focal Plane Y Resolution	3908.141962
Focal Plane Resolution Unit	inches
Custom Rendered	Normal
Exposure Mode	Manual
White Balance	Auto

Scene Capture Type	Standard	
Macro Mode	Normal	
Self Timer	2 s	
Quality	RAW	
Canon Flash Mode	Off	
Continuous Drive	Single	
Focus Mode	Manual Focus (3)	
Record Mode	CR2	
Canon Image Size	n/a	
Easy Mode	Manual	
Digital Zoom	None	
Contrast	Normal	
Saturation	Normal	
Metering Mode	Evaluative	
Focus Range	Not Known	
Canon Exposure Mode	Manual	
Lens Type	Canon EF 100mm f/2.8L Macro IS USM	
Max Focal Length	100 mm	
Min Focal Length	100 mm	
Focal Units	1/mm	
Max Aperture	2.8	
Min Aperture	32	
Flash Activity	0	
Flash Bits	(none)	
Zoom Source Width	0	
Zoom Target Width	0	
Manual Flash Output	n/a	
Color Tone	Normal	
SRAW Quality	n/a	
Focal Length	100 mm Focal Plane X Unknown	52635
Focal Plane Y Unknown	63988	
Canon Flash Info	0 0 0 0	
Auto ISO	100	
Base ISO	100	
Measured EV	6.50	
Target Aperture	9	
Target Exposure Time	1/32	
Exposure Compensation	0	
White Balance	Auto	
Slow Shutter	None	

Shot Number In Continuous Burst	0
Optical Zoom Code	n/a
Camera Temperature	36 C
Flash Guide Number	0
Flash Exposure Compensation	0
Auto Exposure Bracketing	Off
AEB Bracket Value	0
Control Mode	Camera Local Control
F Number	9
Exposure Time	1/32
Measured EV 2	6.5
Bulb Duration	0
Camera Type	EOS High-end
ND Filter	n/a
Canon Image Type	Canon EOS 5D Mark II
Canon Firmware Version	Firmware Version 2.0.4
Owner Name	Marek Bromberek
Serial Number	1921200087
F Number	9.1
Exposure Time	1/32
ISO	100
Highlight Tone Priority	Off
Flash Metering Mode	Off
Camera Temperature	36 C
Focal Length	100 mm
Camera Orientation	Horizontal (normal)
Focus Distance Upper	0.32 m
Focus Distance Lower	0.31 m
White Balance	Auto
Color Temperature	5200
Picture Style	Faithful
High ISO Noise Reduction	Low
Auto Lighting Optimizer	Standard
Lens Type	Canon EF 100mm f/2.8L Macro IS USM
Min Focal Length	100 mm
Max Focal Length	100 mm
Firmware Version	2.0.4
File Index	6868
Directory Index	100
Contrast Standard	0
Sharpness Standard	3

Saturation Standard	0
Color Tone Standard	0
Filter Effect Standard	n/a
Toning Effect Standard	n/a
Contrast Portrait	0
Sharpness Portrait	2
Saturation Portrait	0
Color Tone Portrait	0
Filter Effect Portrait	n/a
Toning Effect Portrait	n/a
Contrast Landscape	0
Sharpness Landscape	4
Saturation Landscape	0
Color Tone Landscape	0
Filter Effect Landscape	n/a
Toning Effect Landscape	n/a
Contrast Neutral	0
Sharpness Neutral	0
Saturation Neutral	0
Color Tone Neutral	0
Filter Effect Neutral	n/a
Toning Effect Neutral	n/a
Contrast Faithful	0
Sharpness Faithful	0
Saturation Faithful	0
Color Tone Faithful	0
Filter Effect Faithful	n/a
Toning Effect Faithful	n/a
Contrast Monochrome	0
Sharpness Monochrome	3
Saturation Monochrome	n/a
Color Tone Monochrome	n/a
Filter Effect Monochrome	None
Toning Effect Monochrome	None
Contrast User Def 1	0
Sharpness User Def 1	3
Saturation User Def 1	0
Color Tone User Def 1	0
Filter Effect User Def 1	None
Toning Effect User Def 1	None
Contrast User Def 2	0

Sharpness User Def 2	3
Saturation User Def 2	0
Color Tone User Def 2	0
Filter Effect User Def 2	None
Toning Effect User Def 2	None
Contrast User Def 3	0
Sharpness User Def 3	3
Saturation User Def 3	0
Color Tone User Def 3	0
Filter Effect User Def 3	None
Toning Effect User Def 3	None
User Def 1 Picture Style	Standard
User Def 2 Picture Style	Standard
User Def 3 Picture Style	Standard
Canon Model ID	EOS 5D Mark II
Thumbnail Image Valid Area	0 159 7 112
Serial Number Format	Format 2
Canon 0x0019	1
AF Info Size	96
AF Area Mode	Off (Manual Focus)
Num AF Points	9
Valid AF Points	9
Canon Image Width	5616
Canon Image Height	3744
AF Image Width	5616
AF Image Height	3744
AF Area Widths	84 84 101 84 84 84 101 84 123
AF Area Heights	101 101 74 101 101 101 74 101 131
AF Area X Positions	-1173 -561 0 561 1173 561 0 -561 0
AF Area Y Positions	0 280 501 280 0 -280 -501 -280 0
AF Points In Focus	(none)
AF Points Selected	8
Original Decision Data Offset	0
Bracket Mode	Off
Bracket Value	0
Bracket Shot Number	0
Raw Jpg Size	Large
Long Exposure Noise Reduction 2	Off
WB Bracket Mode	Off
WB Bracket Value AB	0
WB Bracket Value GM	0

Live View Shooting	Off
Focus Distance Upper	0.32 m
Focus Distance Lower	0.31 m
Flash Exposure Lock	Off
Lens Model	EF100mm f/2.8L Macro IS USM
Internal Serial Number	
Dust Removal Data	(Binary data 1024 bytes, use -b option to extract)
Crop Left Margin	0
Crop Right Margin	0
Crop Top Margin	0
Crop Bottom Margin	0
Aspect Ratio	3:2
Cropped Image Width	5616
Cropped Image Height	3744
Cropped Image Left	0
Cropped Image Top	0
Tone Curve	Standard
Sharpness	0
Sharpness Frequency	n/a
Sensor Red Level	0
Sensor Blue Level	0
White Balance Red	0
White Balance Blue	0
Color Temperature	5200
Picture Style	Faithful
Digital Gain	0
WB Shift AB	0
WB Shift GM	0
Measured RGGB	461 1024 1024 603
Color Space	Adobe RGB
VRD Offset	0
Sensor Width	5792
Sensor Height	3804
Sensor Left Border	168
Sensor Top Border	56
Sensor Right Border	5783
Sensor Bottom Border	3799
Black Mask Left Border	0
Black Mask Top Border	0
Black Mask Right Border	0
Black Mask Bottom Border	0

Color Data Version	6 (50D/5DmkII)
WB RGGB Levels As Shot	2405 1024 1024 1565
Color Temp As Shot	5806
WB RGGB Levels Auto	2405 1024 1024 1565
Color Temp Auto	5806
WB RGGB Levels Measured	2403 1023 1024 1564
Color Temp Measured	5806
WB RGGB Levels Unknown	497 1170 1170 766
Color Temp Unknown	5819
WB RGGB Levels Daylight	2226 1024 1024 1664
Color Temp Daylight	5200
WB RGGB Levels Shade	2551 1024 1024 1378
Color Temp Shade	7000
WB RGGB Levels Cloudy	2394 1024 1024 1502
Color Temp Cloudy	6000
WB RGGB Levels Tungsten	1695 1083 1083 2766
Color Temp Tungsten	3200
WB RGGB Levels Fluorescent	1928 1024 1024 2422
Color Temp Fluorescent	3674
WB RGGB Levels Kelvin	2226 1024 1024 1664
Color Temp Kelvin	5200
WB RGGB Levels Flash	2411 1024 1024 1487
Color Temp Flash	6111
WB RGGB Levels Unknown 2	2226 1024 1024 1664
Color Temp Unknown 2	5200
WB RGGB Levels Unknown 3	2226 1024 1024 1664
Color Temp Unknown 3	5200
WB RGGB Levels Unknown 4	2226 1024 1024 1664 Color Temp Unknown 4 5200
WB RGGB Levels Unknown 5	2226 1024 1024 1664
Color Temp Unknown 5	5200
WB RGGB Levels Unknown 6	2226 1024 1024 1664
Color Temp Unknown 6	5200
WB RGGB Levels Unknown 7	971 1024 1024 956
Color Temp Unknown 7	4270
WB RGGB Levels Unknown 8	971 1024 1024 956
Color Temp Unknown 8	4270
WB RGGB Levels Unknown 9	971 1024 1024 956
Color Temp Unknown 9	4270
WB RGGB Levels Unknown 10	971 1024 1024 956
Color Temp Unknown 10	4270

WB RGGB Levels Unknown 11	971 1024 1024 956
Color Temp Unknown 11	4270
Camera Color Calibration 01	-334 365 908 (10900K)
Camera Color Calibration 02	-311 374 876 (10000K)
Camera Color Calibration 03	-254 396 799 (8300K)
Camera Color Calibration 04	-221 411 761 (7000K)
Camera Color Calibration 05	-167 438 698 (6000K)
Camera Color Calibration 06	-136 452 664 (5600K)
Camera Color Calibration 07	-103 471 630 (5200K)
Camera Color Calibration 08	-57 498 585 (4700K)
Camera Color Calibration 09	0 533 535 (4200K)
Camera Color Calibration 10	56 571 487 (3800K)
Camera Color Calibration 11	105 607 448 (3500K)
Camera Color Calibration 12	167 654 401 (3200K)
Camera Color Calibration 13	210 693 374 (3000K)
Camera Color Calibration 14	260 742 344 (2800K)
Camera Color Calibration 15	371 870 289 (2400K)
Average Black Level	1024 1024 1024 1024
Raw Measured RGGB	90392 196529 196487 123415
Per Channel Black Level	1023 1024 1023 1024
Normal White Level	14800
Specular White Level	15312
Linearity Upper Margin	10000
CRW Param	(Binary data 17040 bytes, use -b option to extract)
Flavor	(Binary data 16448 bytes, use -b option to extract)
Picture Style User Def	Standard; Standard; Standard
Picture Style PC	None; None; None
Custom Picture Style File Name	
Canon 0x4011	[...]
Canon 0x4012	
AF Micro Adj Mode	Disable
AF Micro Adj Value	0
Vignetting Corr Version	0
Peripheral Lighting	Off
Chromatic Aberration Corr	Off
Chromatic Aberration Corr	Off
Peripheral Lighting Value	70
Original Image Width	5616
Original Image Height	3744
Peripheral Lighting Setting	Off
Canon 0x4017	395265 2583691264

Exposure Level Increments	1/3 Stop
ISO Speed Increments	1/3 Stop
ISO Expansion	On
AEB Auto Cancel	On
AEB Sequence	0,-,+
Safety Shift	Disable
Flash Sync Speed Av	Auto
Long Exposure Noise Reduction	Off
High ISO Noise Reduction	Low
Highlight Tone Priority	Disable
Auto Lighting Optimizer	Standard
Canon Custom Functions 2 0x0205	0
Lens Drive No AF	Focus search on
Lens AF Stop Button	AF stop
AF Point Selection Method	Normal
Superimposed Display	On
AF Assist Beam	Emits
Mirror Lockup	Disable
AF Point Area Expansion	Disable
AF Microadjustment	Disable; 0; 0; 0; 0
Shutter Button AF On Button	Metering + AF start
AF On AE Lock Button Switch	Disable
Set Button When Shooting	Normal (disabled)
Dial Direction Tv Av	Normal
Focusing Screen	Eg-A
Add Original Decision Data	Off
Assign Func Button	LCD brightness
Interoperability Index	R98 - DCF basic file (sRGB)
Interoperability Version	0100
GPS Version ID	2.2.0.0
Thumbnail Offset	44460
Thumbnail Length	9236
Thumbnail Image	(Binary data 9236 bytes, use -b option to extract)
Image Width	362
Image Height	234
Bits Per Sample	16 16 16
Compression	Uncompressed
Photometric Interpretation	RGB
Strip Offsets	1261892
Samples Per Pixel	3
Rows Per Strip	234
Strip Byte Counts	508248
Planar Configuration	Chunky

Exif 0xc5d9	2
SRaw Type	3
Exif 0xc6dc	348 232 13 2
Compression	JPEG (old-style)
Strip Offsets	2138436
Strip Byte Counts	23245411
Exif 0xc5d8	1
CR2 CFA Pattern	[Green,Blue][Red,Green]
Raw Image Segmentation	2 1936 1920
SRaw Type	1
Aperture	9.0
Drive Mode	Self-timer Operation
File Number	100-6868
ISO	100
Image Size	5616x3744
Lens	100.0 mm
Lens ID	Canon EF 100mm f/2.8L Macro IS USM
Megapixels	21.0
Scale Factor To 35 mm Equivalent	1.0
Shooting Mode	Manual
Shutter Speed	1/30
Create Date	2019:03:22 10:54:46.18
Date/Time Original	2019:03:22 10:54:46.18
Modify Date	2019:03:22 10:54:46.18
WB RGGB Levels	2405 1024 1024 1565
Blue Balance	1.52832
Circle Of Confusion	0.031 mm
Depth Of Field	0.004 m (0.313 - 0.317 m)
Field Of View	20.9 deg
Focal Length	100.0 mm (35 mm equivalent: 97.6 mm)
Hyperfocal Distance	36.09 m
Lens	100.0 mm (35 mm equivalent: 97.6 mm)
Light Value	11.2
Red Balance	2.348633