A Thesis

entitled

Development of Gray Level Co-occurrence Matrix based Support Vector Machines for Particulate Matter Characterization

by

Karpagam Tirumazhisai Manivannan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Engineering

Dr. Vijay Devabhaktuni, Committee Chair

Dr. Ashok Kumar, Committee Co-Chair

Dr. Srinivasa Vemuru, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

August 2012
An Abstract of

Development of Gray Level Co-occurrence Matrix based Support Vector Machines for Particulate Matter Characterization.

by

Karpagam Tirumazhisai Manivannan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Engineering

The University of Toledo
August 2012

Particulate matter is a highly complex mixture of very minute particles and liquid droplets present in the atmosphere. These inhalable particles can potentially cause heart and lungs in human. The destructive effect of these particles is directly linked to their size. Hence, scientific studies are conducted to determine the physical characterization of particulate matter in addition to their chemical composition.

This thesis proposes an efficient tool for physical characterization of particulate matter using Image processing technique. The selection of optimal segmentation algorithm is essential to characterize the particulate matter image captured from Scanning Electron Microscope (SEM). Further, the morphological properties of the particles are estimated from the segmented image. The Support Vector Machine (SVM) is incorporated for automatic selection of optimal segmentation algorithm for the PM image without any human intervention. The Gray Level Co-occurrence Matrix (GLCM) is used as the superior feature descriptor for the classifier to achieve exceptional performance in predicting the best segmentation algorithm for cumulative number of images. In addition,
the classification performance of Support vector machine is compared with Artificial Neural Network (ANN) in terms of prediction accuracy.
For my parents, brother, and friends
Acknowledgements

I wish to express my deepest gratitude to my advisor Dr. Vijay Devabhaktuni, co-advisor Dr. Ashok Kumar and Dr Priyanka Aggarwal for their continued support, guidance and encouragement, without which it would not have been possible to succeed in my research. The financial support from the EECS Department and the Department of Civil Engineering in the form of a graduate/research assistantship is gratefully acknowledged. I am grateful to the assistance from Abhishek Bhat, in terms of providing the Particulate matter images.

Special thanks to all my friends for their enthusiastic and generous support through my research study.

Above all, I wish to express my deep appreciation to my family, and friends who with their love and encouragement have made all this possible.
# Table of Contents

Abstract .................................................................................................................................................. iii

Acknowledgements .................................................................................................................................... v

Table of Contents ................................................................................................................................. vi

List of Tables ........................................................................................................................................... viii

List of Figures .......................................................................................................................................... x

List of Abbreviations .............................................................................................................................. xiii

1. Introduction ......................................................................................................................................... 1

   1.1 Problem Statement ........................................................................................................................ 1

   1.2 Proposed Research Approach .................................................................................................... 2

   1.3 Organization .............................................................................................................................. 3

   1.4 Contribution .............................................................................................................................. 3

2. Literature Review ................................................................................................................................. 4

   2.1 Introduction ............................................................................................................................... 4

   2.2 Overview of Image processing techniques ............................................................................... 5

   2.3 Prior Art ................................................................................................................................... 8

   2.4 Shortcomings of current techniques ........................................................................................ 10

3. Image analyzing techniques .............................................................................................................. 11
List of Tables

Table 5-1: SVM prediction accuracy using histogram and GLCM for three segmentation algorithms. ...................................................................................................... 45

Table 5-2: SVM prediction accuracy using histogram and GLCM for four segmentation algorithms. ...................................................................................................... 46

Table 5-3: SVM prediction accuracy using histogram and GLCM for five segmentation algorithms. ...................................................................................................... 49

Table 5-4: The measurement of morphological parameter such as area, perimeter, aspect ratio and form factor of the particles present in the PM image in Figure 5.8. 52

Table 5-5: Confusion matrix of GLCM-based SVM classifier for Case 3 with 25 training and 125 testing images .................................................................................... 53

Table 5-6: Confusion matrix of GLCM-based SVM classifier for Case 3 with 50 training and 100 testing images ................................................................................... 53

Table 5-7: Confusion matrix of GLCM-based SVM classifier for Case 3 with 75 training and 75 testing images ..................................................................................... 53

Table 5-8: Confusion matrix of GLCM-based SVM classifier for Case 3 with 100 training and 50 testing images. ................................................................. 54

Table 5-9: Confusion matrix of GLCM-based SVM classifier for Case 3 with 125 training and 25 testing images. ................................................................. 54
Table 5-10: Confusion matrix of histogram-based SVM classifier for Case 3 with 25 training and 125 testing images. ................................................................. 54

Table 5-11: Confusion matrix of histogram-based SVM classifier for Case 3 with 50 training and 100 testing images. ................................................................. 54

Table 5-12: Confusion matrix of histogram-based SVM classifier for Case 3 with 75 training and 75 testing images. ................................................................. 55

Table 5-13: Confusion matrix of histogram-based SVM classifier for Case 3 with 100 training and 50 testing images. ................................................................. 55

Table 5-14: Confusion matrix of histogram-based SVM classifier for Case 3 with 25 training and 125 testing images. ................................................................. 55

Table 5-15: Performance of GLCM-based and histogram-based SVM classifiers showing precision obtained from confusion matrix. ........................................... 56

Table 5-16: Performance of GLCM-based and histogram-based SVM classifiers showing sensitivity obtained from confusion matrix. ........................................... 56

Table 5-17: Performance of GLCM-based and histogram-based SVM classifiers showing specificity obtained from confusion matrix. ........................................... 57

Table 5-18: Performance of GLCM-based and histogram-based SVM classifiers showing sensitivity obtained from confusion matrix. ........................................... 57

Table 5.19: Performance comparison of SVM and ANN classifiers in predicting optimal segmentation algorithm................................................................. 58

Table 5-20: Statistical assessment of SVM and ANN methods derived from the confusion matrix. ........................................................................................................ 58

Table A-1: National Ambient Air Quality Standards for Particulate matter .................. 76
List of Figures

Figure 3-1: Segmentation of the PM image applying two algorithms (a) Original image,
(b) Rosin threshold, (c) Otsu threshold.......................................................... 13

Figure 3-2: Segmentation of the PM image applying two algorithms (a) Original image,
(b) Rosin threshold, (c) Otsu threshold.......................................................... 13

Figure 3-3: The intensity histogram of an image containing the PM (a) Particle image, (b)
Histogram....................................................................................................... 18

Figure 3-4: Comparison of different images showing the identical histogram (a) and (c)
The gray level matrix of the image with size 4x4, (b) and (d) Histogram...... 19

Figure 3-5: Resolution cells and their relationship with neighboring pixels at various
angles. ............................................................................................................. 20

Figure 3-6: Image gray-tone spatial dependence matrix with levels 0 to 3 ................. 21

Figure 3-7: Co-occurrence matrices derived for Figure 3.6 at various angles: (a) \( C_H = 0^0 \),
(b) \( C_{RD} = 45^0 \), (c) \( C_V = 90^0 \), (d) \( C_{LD} = 135^0 \) ......................................................... 22

Figure 3-8: Run length matrices calculated for Figure 3.6 at various angles: (a) \( R_H = 0^0 \)
(b) \( R_{RD} = 45^0 \), (c) \( R_V = 90^0 \), (d) \( R_{LD} = 135^0 \) ......................................................... 24

Figure 4-1: Classification of objects using linear function........................................ 28

Figure 4-2: Classification of objects using complex kernel function ..................... 29

Figure 4-3: Schematic of a hyperplane seperating two classes............................... 30
Figure 4-4: Flow-chart showing GLCM-based and histogram-based SVM training. ...... 35

Figure 4-5: GLCM-based ANN model using MLP network. ........................................... 37

Figure 5-1: Comparison of SVM predicted algorithm with other algorithms: (a) Original
image, (b) Minimum error threshold (SVM prediction by GLCM), (c) Kapur
threshold, (d) Rosin threshold (SVM prediction using histogram). ...................... 44

Figure 5-2: Comparison of SVM predicted algorithm with other algorithms: (a) Original
image, (b) Minimum error threshold (SVM prediction using histogram), (c)
Sobel method (SVM prediction by GLCM), (d) Otsu threshold, (e) Kapur
threshold.................................................................................................................. 46

Figure 5-3: SVM prediction accuracy using histogram and GLCM for four segmentation
algorithms. ................................................................................................................. 47

Figure 5-4: Comparison of SVM predicted algorithm with other algorithms: (a) Original
image, (b) Kapur threshold, (c) Minimum error threshold, (d) Otsu threshold
(SVM prediction by GLCM), (e) Rosin threshold, (f) Sobel method (SVM
prediction using histogram). ................................................................................... 48

Figure 5-5: Histogram accuracy for different training and testing images applying three,
four and five sets of segmentation algorithms................................................. 49

Figure 5-6: GLCM accuracy for different training and testing images applying three, four
and five sets of segmentation algorithms............................................................... 50

Figure 5-7: Number of particles present in the image before and after segmentation..... 51

Figure 5-8: The optimal segmented image showing the number of particles: (a) Original
image (b) segmented using otsu algorithm.............................................................. 52

Figure A-1: PM2.5 emission (Tons per square mile) ............................................. 73
Figure A-2: PM10 emission (Tons per square mile) ........................................................ 74
Figure A-3: PM2.5 emission by source sector ................................................................. 75
Figure A-4: PM10 emission by source sector ................................................................. 75
Figure A-5: Comparison of PM2.5 emission in 2001 and 2011 by EPA. ....................... 77
Figure A-6: Comparison of PM2.5 emission in 2001 and 2011 by EPA. ....................... 78
Figure D-1: Schematic diagram showing the image classification by SVM classifier
trained by the features extracted using the first-order histogram and second-order GLCM................................................................. 86
List of Abbreviations

ANN...... Artificial Neural Network
DFA...... Discriminant Function Analysis
GLCM...... Gray Level Co-occurrence Matrix
GLDM...... Gray Level Difference Matrix
GLRLM...... Gray Level Run Length Method
ID......... Inverse Difference
LOO ...... Leave-one out
LPI......... Multi layer Perceptron
PM.......... Particulate Matter
PSM......... Power Spectrum Method
PTFE......... polystyrenefluoroethylene
RBF......... Radial Basis Function
SEM......... Scanning Electron Microscope
SPIP......... Scanning Probe Image Processor
SVM......... Support Vector Machine
Chapter 1

Introduction

1.1 Problem Statement

Particulate matter (PM) contains organic chemicals, acids, metals and soil or dust particles that are emitted from sources such as factories, power plants, vehicles, construction activity, and natural sources including fires and windblown dust [1]. The PM is regulated in the U.S. by developing human health-based (primary standard) and environmentally-based (secondary standard) criteria for setting permissible levels. The particulate matter can be classified as ultrafine, fine, and coarse particles depending on their size. These particles exist in different shapes and densities in the air, and hence, the aerodynamic diameter has been recognized as a simple means of defining the size of the particles [2]. Fine particles with aerodynamic diameter less than 2.5 µm and greater than 0.1 µm are emitted by toxic organic smelting and processing metals, in addition mainly due to driving automobiles. Coarse particles having the aerodynamic diameter of greater than 2.5µm and less than 10µm are smoke, dirt and dust that are emitted from factories, farming and roads and are blown by wind [3]. Ultrafine particles having aerodynamic diameter less than 0.1 µm are considered to have serious effect on human health. The PM
have longer half-life with higher suspension time in air and travel longer downwind distances. These particles that are inhaled can exert a toxic effect due to their physical or chemical characteristics. The particles carry the absorbed toxic substances causing decrease in functioning of lungs, breathing problems, irregular heartbeat, nonfatal heart attacks and premature death of people suffering from heart or lung diseases. The size of human hair strand is 70-100 µm which is observed to be approximately ten times greater than the PM present in the atmosphere. Further, high specific deposition is noted when inhaled as singlet rather than aggregated particles. Hence, it is vital to determine the number, morphology, size distribution, and chemical composition of these particles so as to reflect their intensity of harmful effects on human health. Such information is useful in refining the risk assessments [4].

1.2 Proposed Research Approach

The physical characterization of the particulate matter is studied by using the images of the filter paper containing the particulate matter that are captured using Scanning Electron Microscope (SEM) at The University of Toledo. The first phase of the thesis presents an efficient and highly reliable automatic selection of optimal segmentation algorithm for the image containing the particulate matter. The prediction of best segmentation algorithm for an image is the essential step for particle characterization since one single segmentation algorithm may not produce the best result for all types of images. The automatic selection process is achieved employing Support vector machine (SVM) trained using the images features extracted by the Gray level co-occurrence matrix (GLCM). The second phase of the thesis emphasizes the comparison of Support
vector machine and Artificial neural network (ANN) classifiers in efficiently predicting the optimal segmentation algorithm for the PM image.

The objective of this thesis is to:

1. Develop an efficient feature extraction technique to train the image classifier in predicting optimal segmentation algorithm to facilitate particle characterization.
2. Present the performance comparison of SVM and ANN classifiers trained by GLCM.

1.3 Organization

The remainder of the document is organized as follows. Chapter 2 discusses the literature review. Chapter 3 describes about the image analyzing techniques including segmentation and feature extraction process. Chapter 4 presents the SVM and ANN classifiers with the detailed procedure involved in the preparation of training and testing dataset for the classifiers. Finally, Chapter 5 gives the conclusions of the work done and future work.

1.4 Contribution

The morphological properties of PM are determined to estimate their harmful effect on human health and environment. The contribution of this thesis is to develop GLCM-based SVM classifier to predict the optimal segmentation algorithm for the PM image to facilitate particle characterization. The proposed technique of training the classifier illustrated better performance than the current techniques of PM characterization.
Chapter 2

Literature Review

2.1 Introduction

A field study was conducted to examine the emission of PM during the summer of 2009. The samples were collected on polytetrafluoroethylene (PTFE) filter papers using GRIMM 1.108 aerosol monitor. The filter papers are then sputter coated with a thin layer (15 nm) of gold to perform SEM analysis. The experiment was conducted on a high vacuum mode for quality images. The filter paper was divided into 1000 µm x 1000 µm grids and roughly around 200 images were captured from each filter samples [5,6]. The images captured were clear and with right contrast. In general, the morphology of PM such as the shape and size distribution of the particles was performed by sieving and microscopy. However, recently image analysis techniques are widely used to determine the physical characterization for cumulative number of images. Several commercial software are available for particle image analysis such as Image-pro [7], Clemex vision PE [8], and PAX-it [9], ImageJ [10] and scanning probe image processor (SPIP) [11]. The drawbacks of these software are their inability to automatically predict the best
segmentation algorithm for a given image. The proposed technique presents the automatic selection of optimal segmentation algorithm employing GLCM-based SVM.

2.2 Overview of Image processing techniques

Segmentation refers to the process of partitioning a digital image into multiple segments so as to separate the particles from the image background [9]. The segmentation of particles is achieved by identifying the pixels in each particle or locating the boundary of the particle. The former technique is based on intensity of pixels, however other attributes such as texture associated with each pixel may also contribute for segmentation. The boundary pixels are identified by using image gradient that possesses high values at the edges of the particle [13]. The main objective of segmentation is to categorize the image into different portions inhibiting homogenous characteristics or features [14]. Numerous techniques are available for segmentation including region-based, thresholding, clustering and edge-based algorithms. Every technique has their own merits and demerits and no specific technique can produce satisfactory result for all the images [15]. Perhaps, the accurate segmentation by one technique may not produce a consistent result for every other image. Once the image is segmented accurately, the morphological diagnosis of the particles can be performed with ease. Accurate and reliable segmentation is an essential step in determining the valuable, quantitative information on size, shape and texture which depicts their harmful effect when inhaled by human [16].

The selection of optimal segmentation for a single image can be detected manually. However, it is time consuming and strenuous to predict the segmentation algorithm for
cumulative number of images captured from SEM. Thus an efficient technique is required to automatically detect the optimal segmentation algorithm from the stack of images instantly without any human intervention [17]. This is possible by using the machine learning technique that learns from the training data fed by the user and creates a model. Further, the created model is applied to predict the output during the testing stage. The training data should effectively describe the image for accurate classification and to output the best segmentation algorithm [18]. Thus the features extracted from the image influences the performance of machine learning technique.

Feature extraction is the process of generating some quantifiable property to enable image classification. The image features can be classified into three types such as color, shape and texture [19]. The color feature is extracted from the image by color histogram or color co-occurrence matrix [20]. The color histogram can be assessed using three different color spaces including RGB, normalized RGB and HIS color spaces. On the other hand, color co-occurrence matrix can be calculated using the four neighboring pixels or eight neighboring pixels present in the image. The shape feature is extracted using two methods; region based and contour based [21]. The region-based method utilizes the entire area of the particle for shape description while the contour based method detects the information present in the contour of the particle. The third type of extracting features using texture is categorized into structural, statistical, model-based and transform-based [22]. The detailed comparisons of these methods are discussed in chapter 3. The statistical method measures the spatial interaction of image pixels and is further classified according to number of pixels defining the local features such as first-order, second-order and higher-order statistics [23]. Histogram based method follows the
first-order statistics and is widely used for feature extraction since they are fast and simple methods. Features derived from this approach include moments such as mean, standard deviation, average energy, etc. [24]. However, in the histogram approach the relationship with the neighboring pixels is not considered, which limits its performance. The GLCM method following the second-order statistics determines the spatial relationship between the pixels by calculating the difference in intensities between the center pixel and its neighbors [25]. Other available methods include gray level run-length method (GLRLM), gray level difference matrix (GLDM) and power spectrum method (PSM). The GLCM-based methods outperformed the other conventional segmentation methods in various applications such as terrain classification, lamb grading and medical image analysis [26-29], mainly due to the utilization of second order probabilities.

There are a number of machine learning techniques for image classification like artificial neural networks (ANN), fuzzy logic, genetic algorithm, and SVMs. The structure of ANNs is difficult to understand and they may fail to capture unique attributes in the training phase. Fuzzy logic requires prior knowledge about the system, while genetic algorithms suffer complications in adequately representing the training/output data [30]. Out of the various available machine learning techniques, SVM is chosen because of its high generalization capability along with no additional knowledge requirement, even with high dimensionality of the input space [31-32]. The performance comparison of SVM and ANN classifiers in predicting the optimal segmentation algorithm is presented in chapter 5.
2.3 Prior Art

The recent trends in texture classification has been discussed by Tou, et al., [33] where the GLCM method is highlighted to be older and simpler among the statistical approach of representing the image. Apparently, signal processing methods has been recently popular for feature extraction and are producing good results with higher accuracy. On the other hand, the computational complexity in implementing these expensive methods has constrained the researchers to abide with the older techniques. However, superior performance is achieved by embedding GLCM with the other statistical techniques. Significant drawbacks of texture classification techniques is illustrated, for instance, the structural method can be implemented only on structured textures which are rarely available while the model based technique admits difficulty in implementation due to the complex parameters involved.

The comparison of statistical features such as GLCM, GLDM and GLRM to facilitate the performance analysis of lamb grading has been presented by Chandraratne [26]. The experimental analysis was performed with 160 images obtained from the lamb chops. Around 148 features (36 GLDM, 90 GLCM, 10 GLRM and 12 geometric) were extracted from each image to train the neural network. Discriminant function analysis (DFA) and ANNs are the classifiers used for texture classification. The results confirmed that the GLCM method outperformed the other feature extraction methods, in addition, DFA classifier obtained an improved performance of 8.8% using GLCM compared to GLDM and GLRM methods. Further, extension of the study combining the features obtained from GLCM, GLDM and GLRM illustrated exceptional results. However, the
combination of features for image classification does not produce promising outcome for all kinds of images.

The automatic classification of texture based on GLCM, GLDM, GLRM and fourier power spectrum is studied Weszka, et al., [27]. The classification performance was conducted on different sets of terrain sample images obtained from LANDSAT. The classification results obtained from the different samples of images depicted that the features extracted from the GLCM produced better classification results compared to the fourier spectrum method and the local statistics. The fourier power spectrum achieved poor classification due to the spurious values obtained from the image whereas the statistical features are modeled more appropriately in the space domain that can capture the essential difference among the samples.

Conners and Harlow [28] made a theoretical comparison of texture algorithms in order to perform the automatic texture discrimination. The relative abilities of the GLCM, GLDM, GLRM and the power spectral method (PSM) were estimated in an attempt to distinguish terrain types from aerial photographs. The study involved the performance comparison of various feature descriptors in efficiently acquiring the texture content information from the image rather focusing on the set of features utilized by each algorithm. The significance of using the spatial distance $d = 1$ for co-occurrence matrix formation (refer sub-section 3.4.4.2 for explanation) in effectively describing the textures is also highlighted. Finally, the results indicate that the GLCM and GLDM are powerful compared to PSM. However, the GLCM can discriminate texture much better than the other methods.
2.4 Shortcomings of current techniques

The first-order method of feature extraction incorporating the image gray levels and count of pixels has the drawback of poor prediction performance during classification due to information lacking on the spatial interaction of pixels intensities [34]. On the other hand, the higher-order feature descriptors such as GLRM, GLDM, fourier spectrum and PSM suffers from the computational complexity and inadequately depicting the image pattern. The performance of the image classifiers depends primarily on the texture content information extracted from the image and their efficient representation of data to train the classifier [35-36].

In this thesis, GLCM is proposed as the superior feature extraction technique that effectively represents the texture pattern by acquiring the spatial interaction of pixels from the image. The feature extraction using GLCM is widely used recently in applications including iris recognition, remote sensing, medical diagnosis, etc. Further, SVM is used as the classification tool for predicting the optimal segmentation algorithm for the image to facilitate the physical characterization of PM present in the environment.
Chapter 3

Image analyzing techniques

3.1 Introduction

A digital image is a discrete two dimensional function $(x,y)$ consisting of $y$ rows and $x$ columns. The resolution of an image is represented as $X \times Y$ where $f(0,0)$ locates the top left corner of the image and the $f(X-1,Y-1)$ to the bottom right corner. Each distinct coordinate in an image is the picture element and is called as pixel while the amplitude of $f$ at any coordinates is called the intensity or gray level of the image [37]. The output of $f(x,y)$ for each pixel depends on the type of image. The PM image obtained from the SEM is a grayscale image that can measure the light intensity only. Each pixel is a scalar proportional to the brightness where the minimum brightness is black and the maximum brightness is white. The brightness in grayscale images are quantized into $L$ levels, where $f(x,y) \epsilon \{0,1,..,L-1\}$ and $L = 2^n$ representing image having $n$ bits per pixel. In general, the grayscale image consists of 8 bits per pixel producing 256 distinct gray levels [38].
3.2 Pre-processing

The PM image captured from SEM undergoes various stages and there is a probability that the random changes are introduced into the values of pixels in the image. These changes are called noise and can be eliminated by using suitable filters [39]. The process of removing unwanted noise or distortion in the image is termed filtering. This enables the image enhancement without destroying the important features of the image. The filters such as mean or median can be applied for this process. The mean filter computes the average of the current pixel and its neighbor unlike the median filter which determines the median value. The median filter attains better results compared to the mean filter by removing the salt and pepper noise from the image [40].

3.3 Segmentation

Segmentation is the process of converting a digital image into semantically interpretable regions by locating the boundaries such as lines and curves of the visible particles [38]. Segmentation divides an image into its constituent regions or objects such that a set of connected pixels have similar properties like color, intensity or texture. In this process, labels are assigned to each image pixel and the pixels with same label are grouped as either particle pixel or background pixel. Various segmentation techniques usually fall under four main categories, namely, clustering, thresholding, edge-detection and region-based methods [41].

The clustering method groups the pixels into clusters and distinguishes them by color or texture. The thresholding method converts multilevel images into binary images (black and white). The edge-detection method locates edges of the object in the image while the
region-based technique distinguishes the individual pixel from the group of pixels [41]. In our study, the thresholding methods like Kapur [42], Rosin [43], Otsu [44], Minimum error [45] and one edge detection method known as Sobel [46] are incorporated to segment the PM images.

Figure 3-1 and Figure 3-2 show the image segmented by Otsu and Rosin thresholding methods. It is apparent that the Rosin method produced good result for first image (Figure 3-1) while for the second image (Figure 3-2), Otsu method gave better results. Hence, it is evident that the same method might not give optimum results for all images.

![Figure 3-1: Segmentation of the PM image applying two algorithms](image1)

(a) Original image, (b) Rosin threshold, (c) Otsu threshold.

![Figure 3-2: Segmentation of the PM image applying two algorithms](image2)

(a) Original image, (b) Rosin threshold, (c) Otsu threshold.
3.3.1 Measurements

The physical characterization of the particle is estimated from the segmented image using the image analysis technique. The size of the particle such as area, perimeter, width, boundary fractal dimension, etc and the shape of the particle including form factor, roundness, convexity, aspect ratio, etc can be determined [47]. The area of the particle is evaluated by counting the number of pixels present in each particle while the pixel count at the contour of the particle measures the perimeter. The aspect ratio is defined as the ratio of maximum diameter of the particle to the minimum diameter (1) while the form factor validates the particle’s similarity to the circle based on the measurement of perimeter as given by the equation (2). The form factor for the circle is one and is measured less than one for any other shapes. The aspect ratio is sensitive to the particle’s elongation, on contrary; the form factor is sensitive to particle’s outline roughness [48].

\[
\text{Aspect ratio} = \frac{\text{Maximum diameter}}{\text{Minimum diameter}} \quad (1)
\]

\[
\text{Form factor} = \frac{4\pi(\text{Area})}{\text{Perimeter}^2} \quad (2)
\]

3.4 Feature extraction

Feature extraction is the measurement of quantifiable property which specifies the significant characteristics of an object [29]. It is the process of locating points that either separates the objects from one another or represents changes in the surface geometry of an object. The extracted features like color, texture and shape are represented as vectors and have a very high influence on the classification efficacy. The feature extraction using
texture analysis is a method that represents the spatial distribution of intensity variations in original image. The feature extraction from an image using texture can be classified as structural, statistical, model-based and transform-based method.

### 3.4.1 Structural

The structural approach represents a texture by definite primitives (microtexture) and a hierarchy of spatial arrangements (macrotexture) of those primitives. The primitive is defined as a single pixel comprising a gray level, however it may be a collection of pixels as well. It consists of specific attributes such as gray tones, local properties, shape measures of the connected region or homogeneity of its local property. The structural methods are suitable for textures containing primitives that are described by wider attributes such as shape description than just tonal properties. The texture of an image is described mainly by the primitives and the placement rules. The placement rule is defined as the probability of the chosen primitive to be placed at a particular location [50].

The structural element is modeled with a set of resolution cells having specific shape such as a line or a square from which a new binary image is created. Further, the textural features are obtained from this binary image by counting the number of resolution cells with the value 1. The structural approach provides a good symbolic description of the image pattern; however, this feature is more useful for the purpose of synthesis than analysis [51]. The other drawback is that the method can be applied only for binary images.
3.4.2 Model-based

The model-based texture analysis is performed by estimating the parameters either by fractal or stochastic models. The stochastic approach leads to computational complexity, on the other hand, the fractal method shows good performance only for natural textures [52]. The objective of this method is to create an empirical model of each pixel in the image based on the weighted average of the neighborhood pixel intensities. Hence, the texture features are described by the parameters estimated from the model.

The major problem in implementing the model-based method is the selection of order for the model by correctly identifying the neighbors. The order of a 2-D autoregressive model can be estimated by multiple partial autocorrelation measure and the Schwarz criterion. However, the experimental results proved that the number of parameters required to describe the texture should be more than 30 which in turn leads to the computational complexity. The accuracy and the parameters estimated are inversely proportional, apparently indicating the decrease in accuracy with increase in number of parameters involved for texture description. The other drawback is that the re-estimation of parameters with the change in order of the model consequently deprives the use of this method for practical applications [53].

3.4.3 Transform-based methods

Transform-based method involves the analysis of texture using frequency achieved using the spatial filter or filtering in frequency domain. The textural characteristics are extracted by converting an image into a new form using the spatial frequency properties of the pixel intensity variations. The features extracted depends on the type of transform used such as fourier transform, laws texture measures, gabor filters, etc that measures the
characteristics of texture based on frequency [54]. The Fourier transform focuses on the global frequency domain lacking information on spatial domain and hence leading to a poor performance. On the other hand, Gabor filter estimates the spatial localization, however the spatial structures in case of natural textures cannot be localized due to unavailability of single filter resolution.

The limitation of transform-based method is that the texture analysis can be estimated for the images containing smooth regions where all the energy is concentrated in the low frequency coefficient and cannot produce desired result for the images containing edges with lot of energy in high frequencies [55].

### 3.4.4 Statistical

Statistical method is the simplest approach which analyzes the spatial distribution of image gray levels by determining the local features of each image pixel and derives a set of statistics from their distribution [56]. The statistical method can be categorized into different classes based on the number of pixels defining the local features, i.e., first-order, second-order and higher-orders. The first-order method evaluates the properties of the individual pixels and ignores their spatial interaction. The second and higher-order methods estimate the pixel properties occurring at different locations relative to each other. Histogram belongs to the first-order approach while GLCM follows the second-order statistics.
3.4.4.1 Histogram

Histogram is a graphical representation containing the tonal distribution of all image pixels. The histogram of a digital image with intensity levels in the range of [0, L-1] is a discrete function with L being the number of gray levels and is given by

\[ h(r_k) = n_k, \]

where \( r_k \) is the kth intensity gray level and \( n_k \) is the number of pixels in the image with intensity \( r_k \). Histogram is made up of bins carrying certain intensity values in each bin. The pixels with identical intensity values are grouped together and the total number of pixels lying in each intensity range is assigned to the respective bins. For example, consider an 8-bit image having the intensity value between 0 and 255 making it 256 bins. Here, the bin 0 contains the number of times a gray level 0 occurs in the image, the bin 1 contains the number of times a gray level 1 occurs in the image and so on [57]. The histogram representation for an image is shown in Figure 3-3.

![Figure 3-3: The intensity histogram of an image containing the PM (a) Particle image, (b) Histogram.](image)
In order to elucidate the limitations of histogram, consider two 4x4 images and their specified intensities as shown in Figure 3-4 (a and c). The histogram obtained from these images is shown in Figure 3-4 (b and d). It is evident that although the images are different, the evaluated histograms are the same; thereby producing contradictory results during the classification based on histogram method.
3.4.4.2 Gray Level Co-occurrence Matrix

GLCM is the method of computing the frequency of pixel pairs having the same grey level in the image. The relationship between the reference pixel and the neighboring pixels is calculated to determine the textural features of the image [58]. In Figure 3-5, the resolution cell 1 and 5 are $0^\circ$ nearest neighbors to cell 0. Similarly, resolution cells 2 and 6, 3 and 7, 4 and 8 are $135^\circ, 90^\circ$ and $45^\circ$ nearest neighbors to cell 0. The co-occurrence matrix is formed by evaluating the count of pixel pairs with gray level $i$ occurring adjacent to pixel with gray level $j$ [59]. The relative frequencies of gray level pixel pairs separated by a distance $d$ in a particular direction $\theta$ forms the displacement vector $(d, \theta)$.

![Figure 3-5: Resolution cells and their relationship with neighboring pixels at various angles.](image)

Let $f: V_y \times H_x \to I$ be an image with dimensions, $V_y = (0,1,2, \ldots, I_y - 1)$, $H_x = (0,1,2, \ldots, I_x - 1)$ having a set of quantized gray-tones $G = (0,1,2, \ldots, L - 1)$. The coordinates of this image $V_y \times H_x$ represent the resolution cells containing the gray levels for each pixel. The texture is assessed by the four closely related measures called angular nearest neighbor gray tone spatial dependence matrices. The content information between
two neighboring cells \{((k,l), (m,n))\} separated by a distance \(d\) is represented by \(C_{ij}\) as illustrated in Figure 3-6. Since the textures involved in the images used for this study are micro-textures, the distance \(d=1\) is chosen in order to extract the detailed textural information. Moreover, the pixels are more likely to correlate with the one that is located closer (i.e., with smaller value of \(d\)) than the pixels that are situated far away. In the figure, the displacement vector is \((1,0^\circ)\) with \(k = m = 1\) and \(l = 1, n = 2\) respectively, where \(I(k,l) = i, I(m,n) = j\). Thus the gray tone spatial dependence matrix is the function of the angular relationship between the neighboring resolution cells and the distance between them [60].

![Image gray-tone spatial dependence matrix with levels 0 to 3.](image)

Figure 3-6: Image gray-tone spatial dependence matrix with levels 0 to 3.

The spatial measurement at different angles with the distance \(d\) is given by (4)-(7).

\[
C(i, j, d, 0^\circ) = \#\{(k,l), (m,n): k - m = 0, \lfloor l - n \rfloor = d\} \tag{4}
\]

\[
C(i, j, d, 45^\circ) = \#\{(k,l), (m,n): k - m = d, l - n = -d\} \tag{5}
\]

or

\[
C(i, j, d, 45^\circ) = \#\{(k,l), (m,n): k - m = -d, l - n = d\}
\]
\[
C(i, j, d, 90^0) = \#\{(k, l), (m, n): |k - m| = d, l - n = 0\} \quad (6)
\]

\[
C(i, j, d, 135^0) = \#\{(k, l), (m, n): k - m = d, l - n = d\} \quad (7)
\]

or

\[
C(i, j, d, 135^0) = \#\{(k, l), (m, n): k - m = -d, l - n = -d\}
\]

where \# represents the count of pixel pairs with same intensity level.

In Figure 3-7(a), the matrix element present in the first row third column (i.e., 2) represents the total number of instances that gray levels \(i = 0\) and \(j = 2\) occurred horizontally adjacent to each other in Figure 3-6. Here, the entry at the coordinates (0, 2) is 2 because there are two pixel pairs of (0, 2) with distance \(d = 1\) at angle \(0^0\) in the Figure 3-6. Similarly, the spatial dependence matrix at various angles \(45^0, 90^0\) and \(135^0\) are shown in Figure 3-7 (b-d).

\[
\begin{pmatrix}
0 & 3 & 2 & 1 \\
3 & 2 & 1 & 2 \\
2 & 1 & 2 & 1 \\
1 & 2 & 1 & 0 \\
\end{pmatrix}
\quad \begin{pmatrix}
0 & 2 & 1 & 1 \\
2 & 2 & 1 & 1 \\
1 & 1 & 2 & 1 \\
1 & 1 & 1 & 0 \\
\end{pmatrix}
\quad \begin{pmatrix}
2 & 1 & 1 & 1 \\
1 & 2 & 4 & 1 \\
1 & 4 & 0 & 2 \\
1 & 1 & 2 & 0 \\
\end{pmatrix}
\quad \begin{pmatrix}
0 & 2 & 2 & 0 \\
2 & 0 & 4 & 0 \\
2 & 4 & 0 & 0 \\
0 & 0 & 0 & 2 \\
\end{pmatrix}
\]

Figure 3-7: Co-occurrence matrices derived for Figure 3-6 at various angles: (a) \(C_H = 0^0\), (b) \(C_{RD} = 45^0\), (c) \(C_V = 90^0\), (d) \(C_{LD} = 135^0\).

Further, the spatial dependence of gray levels can be quantified by calculating the 14 textural features of the co-occurrence matrix constructed by Haralick [61]. These 14 textural parameters are highly correlated and hence only four features (8)-(11) are sufficient for obtaining good image classifications [28, 62].

**Energy:**

Energy measures the occurrence of repeated pixel pairs within the image.
Energy = $\sum \sum C(i,j)^2$ \hspace{1cm} (8)

**Contrast:**

Contrast measures the difference between the maximum and minimum value of a contiguous set of pixels.

$$\text{Contrast} = \sum \sum (i - j)^2 C(i,j)$$ \hspace{1cm} (9)

**Entropy:**

Highly correlated to energy, entropy measures the disorder of the image. Non-uniform texture has a high entropy value.

$$\text{Entropy} = - \sum \sum C(i,j) \log (C(i,j))$$ \hspace{1cm} (10)

**Inverse Difference:**

Inverse difference (ID) measures the smoothness of the image.

$$\text{ID} = \sum \sum \frac{C(i,j)}{1 + (i - j)^2}$$ \hspace{1cm} (11)

**3.4.4.3 Higher order statistical feature**

The gray-level run length method (GLRM) is based on the analysis of higher-order statistical information. In this approach, GLRMs contain the information on the run of a particular gray level, or gray level range, at a particular direction. A gray level run is set of linearly connected pixels containing the same gray level. A coarse texture will therefore be dominated by relatively long runs whereas a fine texture will be populated by much shorter runs. The matrix element $(i, j)$ specifies the number of occurrence of the run length $j$ in a specific direction containing the gray level $i$ [63].
For the image $V_y \times H_x$, the number of runs $r'$ with gray level $i$ and run length $j$ in a direction $\theta$ is denoted by

$$R(i, j) = [r'(i, j|\theta)] \quad (12)$$

In Figure 3-8 (a), the matrix element present in the first row first column (i.e., 4) represents the number of occurrence of the gray level $i = 0$ with run length $j = 1$ in Figure 3-6. The entries at the coordinates $(1,1)$ is 4 because the gray level 0 appears at four pixels with the run length 1 at angle $0^0$ in Figure 3-6. Similarly the run length matrix at various angles $45^0$, $90^0$ and $135^0$ are shown in Figure 3-8 (b-d).

$$
\begin{align*}
\begin{pmatrix}
4 & 0 & 0 & 0 \\
3 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 0 & 0 & 0
\end{pmatrix} & \begin{pmatrix}
4 & 0 & 0 & 0 \\
3 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 0 & 0 & 0
\end{pmatrix} & \begin{pmatrix}
2 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 \\
3 & 0 & 0 & 0
\end{pmatrix} & \begin{pmatrix}
4 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
1 & 1 & 0 & 0
\end{pmatrix}
\end{align*}
$$

(a) (b) (c) (d)

Figure 3-8: Run length matrices calculated for Figure 3-6 at various angles: (a) $R_H = 0^0$, (b) $R_{RD} = 45^0$, (c) $R_V = 90^0$, (d) $R_{LD} = 135^0$.

This is analogous to the GLCM technique (Haralick et al., 1973) as four GLCMs are commonly used to describe texture runs in the directions $0^0, 45^0, 90^0$ and $135^0$ on linearly adjacent pixels.

Based on the preceding definition of the run length matrix, the following features are determined [64].

Short Run emphasis

$$f_{SR} = \frac{1}{TR} \sum_{i=0}^{ix-1} \sum_{j=1}^{iy} \frac{R(i,j)}{j^2} \quad (13)$$

Long run emphasis
GrayUlevel distribution

\[ f_{LR} = \frac{1}{T_R} \sum_{i=0}^{l_x-1} \sum_{j=1}^{l_y} j^2 R(i, j) \]  

(14)

Gray-level distribution

\[ f_{GD} = \frac{1}{T_R} \sum_{i=0}^{l_x-1} \left[ \sum_{j=1}^{l_y} R(i, j) \right]^2 \]  

(15)

\[ T_R = \sum_{i=0}^{l_x-1} \sum_{j=1}^{l_y} R(i, j) \]  

(16)

Where \( l_x \) is the maximum number of gray levels, \( l_y \) is the number of different run lengths in the matrix and \( T_R \) serves as a normalizing factor in each of the run length equations.

Different kinds of feature extraction techniques have been discussed previously. However, the statistical method has been focused in this thesis which analyzes the spatial distribution of image gray levels by determining the local features of each image pixel and deriving a set of statistics from their distribution. The textures in gray scale images are distinguished predominantly using second order moments than using any other statistical approaches [61]. The performance comparisons of first-order statistics and the second-order statistics have been illustrated in chapter 5.
Chapter 4

Image classifiers

4.1 Introduction

The machine learning algorithm is used as the classifiers trained using the image features as dataset and play a vital role in image classification. The classifiers are categorized either as supervised or unsupervised learning algorithms. In supervised learning algorithms, the classes are finite predetermined sets that are labeled and are classified into different groups carrying similar features. Thus a mathematical model is constructed in the training phase and is applied to predict the pattern during the testing phase [65]. In unsupervised algorithm, the classifications are not provided initially and the labels are developed automatically. This algorithm seeks a similarity between a set of data called clusters in order to form a classification group [66]. It is apparent that the supervised learning algorithm classifies better incorporating the additional knowledge obtained during the training process. SVMs are one of the supervised learning algorithms and are considered to be a popular classification tool for pattern recognition. The classification performance of SVM is compared with the other supervised algorithm
known as ANN. The procedure involved in the preparation of training and testing data for the classifiers are also discussed in this chapter.

4.2 Support vector machines

SVMs are supervised learning algorithms developed by Vapnik employed for both classification and regression analysis [67]. SVMs work on statistical learning theory and can produce robust, accurate and effective results with less number of training samples. In general, the standard binary classifier is trained with the set of data belonging to two different categories and the SVM training algorithm builds a training model that predicts the class for the new given data. However, recently the multiclass problems are also solved by decomposing the multiclass into several binary classes to design a multiple binary SVM classifiers. SVM performs structural risk minimization i.e., a classifier is created with minimized VC (Vapnik and Chervonenkis) dimension. Hence, the upper bound of generalization error is predominantly reduced by the low VC dimension [68]. Generalization error is termed as bounds on the error rate of a learning machine on unseen data. These bounds are a function of the training error rate and the terms that measure classifier complexity. To minimize the bounds on the generalization error rate, both the sum of the training error rate and the classifier complexity must be minimized.

4.2.1 Optimal separating hyperplane

The SVMs work on the concept of decision plane that separates a set of objects between two different classes. Consider a sample space containing two different objects such as circles and triangles as shown in the Figure 4-1. SVM defines a boundary
between these two objects such that the circles are separated from the triangles by a line called the optimal hyperplane such that they are able to maximize the margin between two classes of objects. The objects lying on the hyperplane are called support vectors. The classifications between two objects that are easily separable using a simple line are called linear classifiers as shown in Figure 4.1. However, the separation between the margins ($\delta$) should be maximum to reduce the function complexity and minimize the bounds on generalization error. The size of margin is not directly dependent on the dimensionality of the data and the classification performance is observed to be extremely good in spite of high dimensionality. Consequently, the problems faced due to overfitting of high dimensionality data are greatly reduced, hence resolving the term ‘curse of dimensionality’. For these reasons, SVMs are employed for PM image classification and mainly, because they are independent of dimensionality of the training data [69].

Figure 4-1: Classification of objects using linear function
Consider another sample space with the objects as shown in the Figure 4-2. The classification of objects in this case is tricky and demands a complex structure for optimal separation. The separation between the two classes would definitely require a curve which is considered to be a complex function rather than a simple line constructed by the linear function. The principle of SVM is to transform the input space of the original mapped objects having complex structure into a feature space to facilitate linear separation of objects. A set of mathematical functions are involved in this mapping termed kernels and are discussed in sub-section 4.2.2.

Figure 4-2: Classification of objects using complex kernel function

Consider a set of class label \( x_p \in \{-1,+1\} \) and set of random data points \( y \). The decision hyperplane is constructed carrying an intercept term \( b \) and a normal vector perpendicular to the hyperplane called the weight vector \( w \) as given by equation (17),
\[ w \cdot y + b = 0 \] (17)

Hence, to maximize the margin between hyperplane and the data, optimal \( w \) and \( b \) parameters are determined from equation (18) and (19)

\[
w = \sum_{\nu=1}^{u} \alpha_{\nu} x_{\nu} y_{\nu},
\]

and

\[
b = \frac{1}{N_{s}} \sum_{\nu=1}^{N_{s}} (w \cdot y_{\nu} - x_{\nu})
\]

where \( \alpha \) is non-negative lagrange multiplier, \( u \) is the number of training sets and \( N_{s} \) is the number of support vectors.

![Schematic of a hyperplane separating two classes.](image)

Figure 4-3: Schematic of a hyperplane separating two classes.

In this thesis, the objects in the input space are the segmentation algorithms \( s \). In order to classify the algorithms into different classes, SVMs assume \( s = 1 \) as class one and the remaining labels as class two. SVMs then find the maximum margin hyperplane between class one and class two using (17). The value of \( s \) is incremented till it reaches
the maximum number of $s$ classes for a multi-class problem. Thus, SVMs are able to create a training model file by assigning the classes for each training image. During the testing phase, this learnt knowledge is applied to classify the images.

### 4.2.2 Kernel functions

The classification performance of SVM depends mainly on the selection of kernel function $k(y_v, y_u)$ and the regularization parameter $C$. The regularization parameter or penalty factor determines the tradeoff between margin maximization and error minimization. Four commonly used kernel functions are listed below [70].

The Simplest of the kernel function is linear kernel and is derived by the inner product of $y_v$ and $y_u$ given by equation (20)

$$k(y_v, y_u) = y_v^T y_u + c$$

(20)

Polynomial kernel is a non-stationary kernel and is applied for the problems containing normalized training data.

$$k(y_v, y_u) = (\alpha y_v^T y_u + c)^d$$

(21)

where $\alpha$ is slope, polynomial degree $d$ and constant $c$

The Radial basis function (RBF) kernel is given by

$$k(y_v, y_u) = exp(-\gamma ||y_v - y_u||^2)$$

(22)

where $\gamma$ controls the width of the RBF kernel. If the parameter is overestimated, then the exponential will behave linearly and the high dimensional projection will lose its non-linear power. On the other hand, if the parameter is underestimated, then the function will lack regularization and the decision boundary will be highly sensitive to noise in the training data.
The sigmoid function originated from neural network is equivalent to the two-layer perceptron model. The sigmoid kernel function is given by

\[ k(y_v, y_u) = \tanh (\propto y_v^T y_u + c) \]  (23)

where slope \( \propto \) and \( c \) are the adjustable parameters. The common value of \( \propto \) is \( \frac{1}{N} \) where \( N \) is the data dimension.

Among all the kernel parameters, the RBF kernel is a reasonable first choice due to the following advantages [71].

- The classification of samples is possible even though the relation between class labels and attributes are non-linear in high dimensional space.
- Less number of hyper parameters are required compared to other kernels. The number of hyper parameters influences the complexity of model selection.
- The weights and the number of support vectors are automatically selected during the training process.
- The numerical computational complexities are extremely minimum.

Once the kernel model has been selected, the two essential parameters are determined. One of them is the regularization parameter \( C \) which derives the tradeoff between minimizing the training error and minimizing model complexity. The second parameter is the kernel parameter \( \gamma \) that implicitly defines the nonlinear mapping from input space to some high dimensional feature space. Hence appropriate parameters are determined for better performance and are implemented by Cross-validation method.
4.2.3 Cross-validation

The model trained with the efficient kernel obtains an accuracy which does not reproduce the true generalization error during validation and a phenomenon called overfitting occurs due to complex model and as a result attains poor generalization performance despite very small training error. Hence, the performance of the learning algorithm is validated using cross-validation to provide the empirical measure of the generalization performance. Cross validation is the statistical method of evaluating and comparing learning algorithms by dividing the data into two sets called the training set and the validation set. This method can be applied for performance estimation, model selection and tuning the kernel parameters as well [72].

Leave-one out cross validation (LOO) is one of the popular cross validation methods. The procedure for performing the method is such that one of the samples from the training dataset is left out and the learning algorithm is trained on the rest of the samples. The trained model is used to predict the label of the one left out earlier. For \( n \) sequences in the training set, this process is repeated \( n \) times leaving each of the \( n \) sequences out and creating a model from the remaining \( n - 1 \) sequences. LOO procedure obtains an unbiased estimate of expected generalization error. Once the kernel parameters are found, the classification results are evaluated by testing the samples. The simplest measurement is the classification accuracy which is calculated from the number of correctly predicted samples divided by the total number of predicted samples [73].

4.2.4 Training SVM using first-order statistics

The histogram is created by aggregating the pixels having certain intensity values in each of the 256 bins. Different segmentation algorithms are applied to every training
image and the best segmentation algorithm is determined. The training dataset is then prepared containing one "target value" (class label or the segmentation algorithm) and several "attributes" (features) with the following format [74].

\[ D = \left\{ \left( x^1, y^1_{1x256} \right), \left( x^2, y^2_{1x256} \right), \ldots, \left( x^t, y^t_{1x256} \right) \right\} \] (24)

where \( x^1, x^2, \ldots, x^t \) are the segmentation algorithms, \( t \) is the total number of training images and \( y^1, y^2, \ldots, y^t \) are feature vectors for \( t \) training images each with 256 entries. The complete SVM training process is illustrated in Figure 4-4.

4.2.5 Training SVM using second-order statistics

The GLCM is calculated for each image at four different angles, namely, \( 0^\circ, 45^\circ, 90^\circ \) and \( 135^\circ \). The haralick parameters (refer to subsection 3.4.4.2) entropy, energy, contrast and inverse difference are then evaluated for these four angles to create a feature vector of 16 bins for an image as given by

\[ D = \left\{ \left( x^1, z^1_{1x16} \right), \left( x^2, z^2_{1x16} \right), \ldots, \left( x^t, z^t_{1x16} \right) \right\} \] (25)

where \( z^1, z^2, \ldots, z^t \) are the feature vectors each containing 16 bins.
Figure 4-4: Flow-chart showing GLCM-based and histogram-based SVM training.
4.2.6 Testing SVM

The SVM is trained separately using the histogram and GLCM features extracted from the PM image and finally a model is created. The trained model is then applied on the test dataset to predict the optimal segmentation algorithm for the cumulative number of test images. Assuming \( m \) testing images, the histogram and GLCM data is created as given by (26) and (27)

\[
D = \begin{pmatrix}
Y_1^{1x256}, \\
Y_2^{1x256}, \\
\vdots \\
Y_t^{1x256}
\end{pmatrix}
\]

(26)

\[
D = \begin{pmatrix}
z_1^{1x16}, \\
z_2^{1x16}, \\
\vdots \\
z_t^{1x16}
\end{pmatrix}
\]

(27)

The training and testing of PM images for classification using SVM was performed by freely available software (SVM-Light; URL: http://svmlight.joachims.org/) [75]. The preparation of training and test data was performed using the MATLAB package (MATLAB 2009a).

4.3 Artificial neural networks

ANNs are the supervised learning algorithms that replicate some function of the human brain. ANNs are general purpose computing tools that are capable of solving complex non-linear problems. ANNs contain a number of individual units (called neurons) that are inter-connected with their associated weights. These networks learn from the training data by adjusting the network parameters, like weights and biases. A
common type of ANN is a multi layer perceptron (MLP) network with backpropagation learning algorithm [76].

### 4.3.1 Backpropagation algorithm

The backpropagation neural network consists of the processing node which is considered to be the basic element. Each processing node acts as a biological neuron and sums the value of its input. This sum is then passed through an activation function which is considered to be any differential function [77].

![GLCM-based ANN model using MLP network.](image)

The processing nodes are arranged into layers and are fully interconnected to the following layer. However, there is no interconnection between the nodes of the same layer. The backpropagation neural network consists of an input layer that behaves as a distribution structure for the data being presented to the network. The input layer is
followed by one or more processing layer called the hidden layers and the final layer is the output layer. In this study, an MLP network with one hidden layer is trained using the backpropagation algorithm to derive a model for selecting the optimal image segmentation algorithm (Figure 4-5). To derive the input-output functional relationship model ($f$), the MLP network is initially assigned random weights and is then trained with the desired input and output patterns [78]. The relationship to be modeled between the input $a$ and the output $b$ is given by equation (28)

$$b = f(a, w)$$

(28)

where $w$ is the interconnecting weight whose value is determined by the network during the training process. The interconnected weights can be adjusted by numerous algorithms to achieve minimal overall training error in multi-layer networks. The iterative process is used to minimize an error function over the network output and a set of target output obtained from the training data set. The training data consists of a pair of data vectors. One of the vector is the features that has to be learned by the network and other vector is the desired output that has to be produced by the network. The main objective of the training is to minimize the overall difference between the desired output and the actual output of the network [79]. The initial stage of training is the entry of training data to the network which is forwarded through the network to the output units. At this stage, the training error $E(w)$ is calculated which is the difference between the desired and actual output as given in equation (29).

$$E(w) = \frac{1}{2} \sum_{i=1}^{t} [\hat{b}^i - b^i]^2$$

(29)

where $\hat{b}^i$ is the trained output corresponding to the input $a_{i \times 16}^i$, $i$ varies from 1 to $t$ (total number of training images) and $b^i$ is the desired output. This error is then fed back to the
input layer with the weights connecting the units being changed in relation to the magnitude of the error. The process is repeated until the error rate is minimized completely.

4.3.2 Network parameters

The training of neural network involves adjusting the initial parameters that strongly influences the performance of neural network in terms of speed and accuracy. However, choosing these parameters wisely will never guarantee the acceptable solution from neural network [80]. The parameters involved in the network model are discussed below.

*Learning parameter*- The learning rate and momentum values are provided by the user that significantly influences the performance of a network.

*Initial weights*- The network performance is influenced by the initial weight settings of the pre-trained network.

*Number of training iterations*- The degree of generalization is controlled by the number of training iterations. The network trained with large number of iterations may not function well for the test data while separating the classes.

*Number of hidden layers and units*- The capacity of the network to learn and generalize is determined by the selection of hidden layers and units.

*Number of input patterns*- The classification accuracy may be affected by the number of training patterns involved to train the network.

4.3.3 Training and testing ANN model

An ANN model is created such that the training error is minimized and is less than $10^{-2}$ with the maximum of 200 iterations. The training sets for ANN using GLCM
method containing the input feature and the corresponding output segmentation algorithm is shown in (30) and testing sets created to test ANN is given in (31).

\[
A = \{(a_{1x16}^1, b^1), (a_{1x16}^2, b^2), ..., (a_{1x16}^t, b^t)\}^T
\]

\[
A = \{(a_{1x16}^1), (a_{1x16}^2), ..., (a_{1x16}^t)\}^T
\]

In this study, the number of input neurons is set to 16, the number of hidden neurons to 25 and the output neuron to 1. The number of hidden neurons is determined using the trial and error method varying between 5 and 50 until a minimum training error percentage is obtained. Finally the samples are validated to determine the classification accuracy. The training and testing of data set for the PM image classification using ANN model was performed by a CAD tool Neuromodeler (Department of Electronics, Carleton University, Ottawa, Canada) [81].

4.4 Performance comparison of SVM and ANN.

The complexity of the ANN model depends on the number of layers and the number of nodes used in the network while the SVM automatically decides the model architecture. It is observed that the standard backpropagation algorithm scales badly to the problem dealing with large data and consequently, increases the run time. In case of tradition neural network, the empirical risk minimization is applied to reduce the training error where as the SVM implements structural minimization to minimize the generalization error by striking a right balance between the training error and the capacity of the machine. In general, the solution of ANN may tend to fall into the local optimal solution, on the other hand, the SVM ensures for global optimal solution [82]. Above all,
SVM is able to solve complex non-linear problems by adjusting the kernel parameters and thereby mapping the data to a higher dimensional space. A linear solution in the higher dimensional space corresponds to a non-linear solution in the lower dimensional space [83]. Hence, SVM classifier is widely used in many applications due to its speed, accuracy and robustness. The performance comparison of SVM and ANN has been illustrated in sub-section 5.3.
Chapter 5

Results and Discussion

5.1 Introduction

To study the characteristics of PM, the airborne particles are collected on the PTFE filter paper using Grimm 1.108 aerosol spectrometer. FEI Quanta 3D SEM then segmented the filter paper into several grids and captured the images from each grid. Particles appeared as bright spots in these images. Around 150 feasible images are selected from the stack of captured images to prepare a database for training and testing samples. The experimental study is partitioned into two phases. The first phase of the chapter illustrates the performance comparison of GLCM-based SVM and the histogram-based SVM for their prediction accuracy. The second phase of the chapter emphasizes the classification performance of SVM and ANN classifiers trained and tested with the features extracted from PM image using GLCM. The classification accuracy in predicting the optimal segmentation algorithm for the PM image is assessed by the statistical analysis formulated using confusion matrix.
5.2 Classification performance of GLCM-based and histogram-based SVMs

A database containing 150 images was selected and randomly divided into training and testing samples. Three cases are implemented and assessed: Case 1 consists of three segmentation algorithms to choose from, namely, Minimum error threshold, Kapur threshold and Rosin threshold. Similarly, Case 2 consists of four algorithms, namely Minimum error threshold, Sobel, Otsu and Kapur algorithms. For Case 3, the samples are segmented by five segmentation algorithms, namely, Kapur threshold, Rosin threshold, Minimum error threshold, Otsu threshold and Sobel algorithms. Equal number of training images is used for each segmentation algorithm for fair assessment of the trained SVM classifier [84].

Figure 5-1 shows the segmented images after applying three algorithms of Case 1. The SVM predicted the Minimum error threshold as the best segmentation algorithm when trained using GLCM, whereas, Rosin was predicted as the best algorithm when trained using histogram features. This resulted into better performance of GLCM-based SVM method compared to the histogram-based SVM classification. The best segmentation algorithm is manually selected by the expert by comparing the segmented image containing prominent particles with the original image. Note that the image shown in Figure 5-1 (a) is randomly selected from a set of 150 images and the segmented images using different algorithms are shown in Figure 5-1 (b-d). Figure 5-1 (b) is perceived by the expert to be the best segmented image among others, as the particles present in the image are protuberant compared to the other image shown in Figure 5-1 (c) and Figure 5-1 (d).
Figure 5-1: Comparison of SVM predicted algorithm with other algorithms: (a) Original image, (b) Minimum error threshold (SVM prediction by GLCM), (c) Kapur threshold, (d) Rosin threshold (SVM prediction using histogram).

Table 5-1 depicts the number of images correctly and incorrectly predicted by SVM classifiers. For 25 training and 125 testing images, the GLCM-based SVM gave 77.6% accuracy while the histogram-based SVM gave 37.6%. Further, on increasing the training images to 125, GLCM-based SVM gave 96.0% accuracy as opposed to 80.0% by histogram method. This clearly demonstrates that the number of images correctly predicted by GLCM is more as compared to the histogram method. Further, GLCM-based SVM required only 12-40 iterations for training as opposed to the histogram
method which required 120-500 iterations for the same number of samples. For instance, GLCM and histogram needed 12 and 122 iterations respectively for 25 training samples.

Table 5-1: SVM prediction accuracy using histogram and GLCM for three segmentation algorithms.

<table>
<thead>
<tr>
<th>Trained Images</th>
<th>Tested Images</th>
<th>Correctly Predicted Images</th>
<th>Incorrectly Predicted Images</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Histogram</td>
<td>GLCM</td>
<td>Histogram</td>
</tr>
<tr>
<td>25</td>
<td>125</td>
<td>47</td>
<td>97</td>
<td>78</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>64</td>
<td>79</td>
<td>36</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>49</td>
<td>63</td>
<td>26</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>34</td>
<td>47</td>
<td>16</td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>20</td>
<td>24</td>
<td>5</td>
</tr>
</tbody>
</table>

For Case 2 (Figure 5-2), GLCM-based SVM gave 96.0% and histogram-based SVM gave 68.0% accuracy with 125 training and 25 testing images. The segmentation by Sobel algorithm produced much cleaner results for this particular image compared to the other algorithms which was accurately predicted by GLCM-based SVM. From Table 5-2, it is observed that with the increase in the number of training images from 25 to 125, the GLCM method accuracy increased from 76.0% to 96.0%.
Figure 5-2: Comparison of SVM predicted algorithm with other algorithms: (a) Original image, (b) Minimum error threshold (SVM prediction using histogram), (c) Sobel method (SVM prediction by GLCM), (d) Otsu threshold, (e) Kapur threshold.

Table 5-2: SVM prediction accuracy using histogram and GLCM for four segmentation algorithms.

<table>
<thead>
<tr>
<th>Trained Images</th>
<th>Tested Images</th>
<th>Correctly Predicted Images</th>
<th>Incorrectly Predicted Images</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Histogram</td>
<td>GLCM</td>
<td>Histogram</td>
</tr>
<tr>
<td>25</td>
<td>125</td>
<td>44</td>
<td>95</td>
<td>81</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>60</td>
<td>79</td>
<td>40</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>46</td>
<td>63</td>
<td>29</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>33</td>
<td>47</td>
<td>17</td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>25</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>
For Case 3, we incorporated five segmentation algorithms to train histogram-based and GLCM-based SVMs. In this Case (Figure 5-4), GLCM-based SVM yielded 88% accuracy compared to 52% in histogram-based SVM with 125 training and 25 test images. The red circles marked in Figure 5-4 (c & f) indicates the presence of noise and partial segmentation of the particles unlike the image segmented by Otsu algorithm (Figure 5-4 (d)) that depicts the prominent segmentation of particles. Further, from Table 5-3, we can clearly establish that the GLCM-based method is more accurate than the histogram-based method for all the cases. Tables 5-1, 5-2 and 5-3 also illustrates that the accuracy produced by applying three segmentation algorithms is higher than that obtained by five segmentation algorithms.
Figure 5.4: Comparison of SVM predicted algorithm with other algorithms: (a) Original image, (b) Kapur threshold, (c) Minimum error threshold, (d) Otsu threshold (SVM prediction by GLCM), (e) Rosin threshold, (f) Sobel method (SVM prediction using histogram).
Table 5-3: SVM prediction accuracy using histogram and GLCM for five segmentation algorithms.

<table>
<thead>
<tr>
<th>Trained Images</th>
<th>Tested Images</th>
<th>Correctly Predicted Images</th>
<th>Incorrectly Predicted Images</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Histogram</td>
<td>GLCM</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>125</td>
<td>41</td>
<td>94</td>
<td>32.8</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>59</td>
<td>75</td>
<td>59.0</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>45</td>
<td>59</td>
<td>60.0</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>32</td>
<td>42</td>
<td>64.0</td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>13</td>
<td>22</td>
<td>52.0</td>
</tr>
</tbody>
</table>

Further, the graph plotted applying different number of segmentation algorithms using histogram (Figure 5-5) and GLCM method (Figure 5-6) prove that the precision can be improved by increasing the number of training images. Thus the accuracy of 96% is achieved by training 125 images and testing 25 images with three and four segmentation algorithms as shown in Table 5-1 and Table 5-2.
Figure 5-6: GLCM accuracy for different training and testing images applying three, four and five sets of segmentation algorithms.

Figure 5-7, compares the number of particles by each of the segmentation algorithm with the actual number of particles present in the image (Figure 5-4) determined using MATLAB. The Otsu algorithm which is predicted to be the best algorithm by GLCM-based SVM identified 36 particles which is closest to the actual 48 number of particles present in the original image. The Sobel algorithm predicted by histogram-based SVM gave 74 particles. However the Otsu algorithm shows lesser number of particles compared to the actual number, the particles segmented by this method seems to be prominent and noise-free compared to the output obtained using other methods and hence considered to be the better segmentation algorithm.
The physical characterization of PM can be further determined from the optimal segmented image. The size and shape parameter such as area, perimeter, aspect ratio and form factor of the particles are calculated [85]. Figure 5-8 shows the original image and the segmented image using Otsu algorithm. The number of particles actually present in the original image is determined as 26 while the particles present in the segmented image is 15. However, the particles are accurately segmented by Otsu algorithm and hence it is predicted as best by SVM. The morphological measurement of the particles is performed and is tabulated in Table 5-4.
Figure 5-8: The optimal segmented image showing the number of particles: (a) Original image (b) segmented using otsu algorithm.

Table 5-4: The measurement of morphological parameter such as area, perimeter, aspect ratio and form factor of the particles present in the PM image in Figure 5-8.

<table>
<thead>
<tr>
<th>Particle</th>
<th>Area</th>
<th>Perimeter</th>
<th>Aspect ratio</th>
<th>Form factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>151.04</td>
<td>70.80155</td>
<td>1.051223</td>
<td>0.378631</td>
</tr>
<tr>
<td>2</td>
<td>1569.28</td>
<td>405.3143</td>
<td>3.30503</td>
<td>0.12004</td>
</tr>
<tr>
<td>3</td>
<td>1064.96</td>
<td>226.6289</td>
<td>1.887044</td>
<td>0.260563</td>
</tr>
<tr>
<td>4</td>
<td>596.48</td>
<td>147.9654</td>
<td>1.41241</td>
<td>0.342362</td>
</tr>
<tr>
<td>5</td>
<td>263.68</td>
<td>144.0877</td>
<td>1.567741</td>
<td>0.1596</td>
</tr>
<tr>
<td>6</td>
<td>399.36</td>
<td>264.6962</td>
<td>1.97166</td>
<td>0.071627</td>
</tr>
<tr>
<td>7</td>
<td>130.56</td>
<td>91.6454</td>
<td>1.713722</td>
<td>0.195343</td>
</tr>
<tr>
<td>8</td>
<td>5478.4</td>
<td>834.6849</td>
<td>5.94303</td>
<td>0.098814</td>
</tr>
<tr>
<td>9</td>
<td>6901.76</td>
<td>625.4302</td>
<td>2.674435</td>
<td>0.221724</td>
</tr>
<tr>
<td>10</td>
<td>222.72</td>
<td>85.28309</td>
<td>1.447801</td>
<td>0.384807</td>
</tr>
<tr>
<td>11</td>
<td>307.2</td>
<td>100.6431</td>
<td>1.254869</td>
<td>0.381121</td>
</tr>
<tr>
<td>12</td>
<td>133.12</td>
<td>62.06116</td>
<td>1.230825</td>
<td>0.434324</td>
</tr>
<tr>
<td>13</td>
<td>83471.36</td>
<td>2068.792</td>
<td>1.624907</td>
<td>0.245083</td>
</tr>
<tr>
<td>14</td>
<td>168.96</td>
<td>72.92232</td>
<td>1.102171</td>
<td>0.399276</td>
</tr>
<tr>
<td>15</td>
<td>624.64</td>
<td>150.9646</td>
<td>1.685721</td>
<td>0.344421</td>
</tr>
</tbody>
</table>

5.2.1 Statistical assessment of GLCM-based and histogram-based methods

The statistical performance analysis of the GLCM-based and the histogram-based methods are evaluated by a confusion matrix method [86]. A confusion matrix displays
the number of correct and incorrect predictions by the classifier compared with the actual classifications in the test data. The matrix arrangement is in such a way that the instances of predicted classes form the columns and the numbers of actual classes are the rows of the matrix. Consider Table 5-5 showing the confusion matrix of GLCM-based SVM for Case 3 with 125 testing images. The entry in row one, column one illustrates that for twenty images, Otsu was correctly predicted by GLCM-based SVM. Similarly, the entry in row two, column one indicates that Rosin algorithm was wrongly predicted as Otsu algorithm for two images by GLCM-based SVM.

Table 5-5: Confusion matrix of GLCM-based SVM classifier for Case 3 with 25 training and 125 testing images

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>Minimum error</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5-6: Confusion matrix of GLCM-based SVM classifier for Case 3 with 50 training and 100 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>2</td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>Minimum error</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 5-7: Confusion matrix of GLCM-based SVM classifier for Case 3 with 75 training and 75 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 5-8: Confusion matrix of GLCM-based SVM classifier for Case 3 with 100 training and 50 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Minimum error</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 5-9: Confusion matrix of GLCM-based SVM classifier for Case 3 with 125 training and 25 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Minimum error</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5-10: Confusion matrix of histogram-based SVM classifier for Case 3 with 25 training and 125 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>11</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Rosin</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>3</td>
<td>0</td>
<td>13</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Minimum error</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>6</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 5-11: Confusion matrix of histogram-based SVM classifier for Case 3 with 50 training and 100 testing images.

<table>
<thead>
<tr>
<th></th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>19</td>
<td>0</td>
</tr>
</tbody>
</table>

54
Table 5-12: Confusion matrix of histogram-based SVM classifier for Case 3 with 75 training and 75 testing images.

<table>
<thead>
<tr>
<th>Minimum error</th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Sobel</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Minimum error</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5-13: Confusion matrix of histogram-based SVM classifier for Case 3 with 100 training and 50 testing images.

<table>
<thead>
<tr>
<th>Minimum error</th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Minimum error</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5-14: Confusion matrix of histogram-based SVM classifier for Case 3 with 25 training and 125 testing images.

<table>
<thead>
<tr>
<th>Minimum error</th>
<th>Otsu</th>
<th>Rosin</th>
<th>Kapur</th>
<th>Sobel</th>
<th>Minimum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otsu</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rosin</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Kapur</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sobel</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Minimum error</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

Table (5-15 to 5-18), tabulates the specificity, sensitivity, precision and F-measure computed using the confusion matrix evaluated from Table (5-5 to 5-14). The sensitivity or recall is the correctly classified proportion of the test samples while the specificity gives the proportion of incorrectly classified test samples. The precision is the ratio of correctly classified images by a single segmentation algorithm among the total number of
correct predictions by all the algorithms whereas the F-measure is the harmonic mean of precision and recall given by (32) [87].

\[
F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{32}
\]

The sensitivity of histogram-based method was 0.35 and 0.80 for 125 and 25 testing images, respectively, while that of GLCM-based method was 0.78 and 0.97, respectively. Similarly, the specificity for histogram method varied from 0.83 to 0.93 while that of GLCM method was 0.94 to 0.99. Further, for GLCM method the largest F-measure is determined to be 0.94 while for histogram it is 0.77. Hence, we conclude that the GLCM-based SVM approach is more effective in selecting the better segmentation algorithm than the histogram-based SVM method.

Table 5-15: Performance of GLCM-based and histogram-based SVM classifiers showing precision obtained from confusion matrix.

<table>
<thead>
<tr>
<th>Trained images</th>
<th>Tested images</th>
<th>Precision</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>125</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>0.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>0.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>0.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5-16: Performance of GLCM-based and histogram-based SVM classifiers showing sensitivity obtained from confusion matrix.

<table>
<thead>
<tr>
<th>Trained images</th>
<th>Tested images</th>
<th>Sensitivity</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>125</td>
<td>0.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>0.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>0.72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>0.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>0.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5-17: Performance of GLCM-based and histogram-based SVM classifiers showing specificity obtained from confusion matrix.

<table>
<thead>
<tr>
<th>Trained images</th>
<th>Tested images</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Histogram</td>
</tr>
<tr>
<td>25</td>
<td>125</td>
<td>0.83</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>0.87</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>0.89</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>0.91</td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Table 5-18: Performance of GLCM-based and histogram-based SVM classifiers showing sensitivity obtained from confusion matrix.

<table>
<thead>
<tr>
<th>Trained images</th>
<th>Tested images</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Histogram</td>
</tr>
<tr>
<td>25</td>
<td>125</td>
<td>0.36</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>0.62</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>0.73</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>0.73</td>
</tr>
<tr>
<td>125</td>
<td>25</td>
<td>0.77</td>
</tr>
</tbody>
</table>

5.3 Performance comparison of GLCM-based SVM and ANN classifiers

The performance of two GLCM-based classifiers, namely, ANN and SVM were assessed by using the same number of training/testing images. A database containing 60 images was randomly selected and divided into 50 training and 10 testing samples. It is evident from Table 5-19 that the ANN gave classification efficiency of 70% for Case 1, 50% for Case 2 and 40% for Case 3 while GLCM-based SVM gave accuracy of 90%, 80% and 80%, respectively. Also, the SVM classifier required only 20 iterations while ANN required 200 iterations for the same number of training images. For Case 3, when the training images were reduced to 25, the accuracy of the SVM dropped to 80% which is still higher than that of ANN classifier (70%). This demonstrates that SVM classifiers perform better even with less number of training samples and overall are more accurate and robust than the ANN classifiers.
Table 5-19: Performance comparison of SVM and ANN classifiers in predicting optimal segmentation algorithm.

<table>
<thead>
<tr>
<th>Trained images</th>
<th>Test images</th>
<th>Accuracy (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANN</td>
<td>SVM</td>
<td>ANN</td>
<td>SVM</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>70</td>
<td>90</td>
<td>50</td>
<td>80</td>
</tr>
</tbody>
</table>

The performance analysis of SVM and ANN methods are determined using the confusion matrix for Case 1, Case 2 and Case 3. Table 5-20 show the specificity, sensitivity, precision and F-measure computed using the confusion matrix for ten testing images. For Case 1, the sensitivity of SVM and ANN classifiers was determined as 0.93 and 0.64 respectively while the specificity was 0.96 and 0.83 respectively. Further, the largest F-measure determined by SVM is 0.90 while ANN gave 0.66 value for Case 1. Hence, our proposed method outperforms ANN method.

Table 5-20: Statistical assessment of SVM and ANN methods derived from the confusion matrix.

<table>
<thead>
<tr>
<th>Case</th>
<th>Precision</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANN</td>
<td>SVM</td>
<td>ANN</td>
<td>SVM</td>
</tr>
<tr>
<td>1</td>
<td>0.68</td>
<td>0.88</td>
<td>0.64</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
<td>0.83</td>
<td>0.45</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>0.36</td>
<td>0.83</td>
<td>0.45</td>
<td>0.80</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis demonstrates an enhanced performance in predicting the optimal segmentation algorithm for the PM images captured using SEM. The images are segmented using the widely applied threshold and edge detection techniques such as Kapur threshold, Rosin threshold, Minimum error threshold, Otsu threshold and Sobel algorithms. The selection of best segmentation algorithm from these algorithms is necessary since the image segmented accurately by one algorithm may not produce the same result for all the images. The SVM which is considered to be an efficient classifier for pattern recognition is employed to predict the optimal segmentation algorithm for the images. The image features are extracted and represented effectively using the GLCM in order to train the SVM. The four highly correlated texture parameters constructed by Haralick such as energy, contrast, entropy and inverse difference are determined from the GLCM at four different angles. Thus GLCM-based SVM achieved an accuracy of 96% in comparison to the histogram-based SVM with the accuracy of 80% when trained with three segmentation algorithms. The SVM trained using the GLCM illustrated excellent
performance due to the additional knowledge extracted from the spatial relations in an image for better classification than using the image histogram alone. Perhaps, the statistical assessment using the confusion matrix states that the GLCM outperformed histogram method. It is evident that the feature descriptors greatly influence the classification accuracy.

The second phase of the thesis demonstrated the performance comparison of machine learning algorithms for the classification of PM image. The SVM trained using GLCM is compared with the popular classifier ANN that is trained using the same features as SVMs. It is observed that the prediction accuracy of the SVM is higher than the ANN. One of the reasons is that the training procedure of SVM tends to converge the global minimum of the objective function unlike the ANN which suffers from existence of local minima. Moreover, choosing the parameters for SVM is less complex than ANN since ANN involves the pre-determined selection of parameters such as learning parameter, initial weights, number of training iterations, number of hidden layers and number of input patterns. Perhaps, increase in number of hidden nodes in the network structure exponentially increases the computation speed. Overall, SVM is always better than ANN in image classification and there is a growing trend of preference for SVMs over ANN in the learning task.

6.2 Future Work

Future work in this direction could include the pre-processing stage for optimal selection of Haralick features inhibiting better textural description of the images. Perhaps,
the exploration of advanced feature extraction technique to train the SVM can achieve higher classification accuracy in predicting the optimal segmentation algorithm for PM images.


References


[81] NeuroModeler, Department of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, K1S5B6, Ontario, Canada, 2011 November, 20.


Appendix – A

Particulate matter emission in Ohio state (2008)

Figure A-1: PM2.5 emission (Tons per square mile)
Figure A-2: PM10 emission (Tons per square mile)
Figure A-3: PM2.5 emission by source sector

Figure A-4: PM10 emission by source sector
Table A-1: National Ambient Air Quality Standards for Particulate matter

<table>
<thead>
<tr>
<th>Pollutant</th>
<th>Primary/Secondary</th>
<th>Averaging time</th>
<th>Level</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PM_{10}$</td>
<td>Primary and secondary</td>
<td>Annual</td>
<td>revoked</td>
<td>Due to a lack of evidence linking health problems with long-term exposure to coarse particle pollution, the agency revoked the annual PM10 standard in 2006.</td>
</tr>
<tr>
<td></td>
<td>24-hour</td>
<td>150μg/m³</td>
<td>Should not exceed more than once per year</td>
<td></td>
</tr>
<tr>
<td>$PM_{2.5}$</td>
<td>Primary and secondary</td>
<td>Annual</td>
<td>15.0μg/m³</td>
<td>The annual arithmetic mean $PM_{2.5}$ within single or multiple community monitored over a period of 3 years should not exceed 15.0μg/m³</td>
</tr>
<tr>
<td></td>
<td>24-hour</td>
<td>35μg/m³</td>
<td>98th percentile of 24-hour concentration within an area monitored over a period of 3 years should not exceed 35μg/m³</td>
<td></td>
</tr>
</tbody>
</table>
Figure A-5: Comparison of PM2.5 emission in 2001 and 2011 by EPA.
Figure A-6: Comparison of PM2.5 emission in 2001 and 2011 by EPA.
Appendix –B

Source code for feature extraction and physical characterization

B.1 Source code to determine GLCM and Haralick features.

clear;
clc;
I1 = imread('app_G1_start_100.jpg');
I2 = medfilt2(I1,[3 3]);
I3 = medfilt2(I2,[3 3]);
I4 = medfilt2(I3,[3 3]);
I5 = medfilt2(I4,[3 3]);
image = medfilt2(I5,[3 3]);

% co-occurrence matrix calculation at different angles
% glcms1 = graycomatrix(image,'Numlevels',8,'Offset',[0 1]);
% glcms2 = graycomatrix(image,'Numlevels',8,'Offset',[-1 -1]);
% glcms3 = graycomatrix(image,'Numlevels',8,'Offset',[-1 0]);
% glcms4 = graycomatrix(image,'Numlevels',8,'Offset',[-1 1]);

% calculation of Inverse difference
[x1,y1]=size(glcms1);
for x1=1:8;
    for y1=1:8;
        feature(1)= glcms1(x1,y1)/((1+(x1-y1)^2));
    end;
end;
[x2,y2]=size(glcms2);
for x2=1:8;
    for y2=1:8;
        feature(2)= glcms2(x2,y2)/((1+(x2-y2)^2));
    end;
end;
[x3,y3]=size(glcms3);
for x3=1:8;
    for y3=1:8;
        feature(3)= glcms3(x3,y3)/((1+(x3-y3)^2));
    end;
end;
end
end
[x4,y4]=size(glcms4);
for x4=1:8;
    for y4=1:8;
        feature(4)= glcms4(x4,y4)/((1+(x4-y4)^2));
    end
end

%calculation of entropy
feature(5)=(entropy(glcms1));
feature(6)=(entropy(glcms2));
feature(7)=(entropy(glcms3));
feature(8)=(entropy(glcms4));

%calculation of contrast

c11 = graycoprops(glcms1,'contrast');
feature(9)=c11.Contrast;
c21 = graycoprops(glcms2,'contrast');
feature(10)=c21.Contrast;
c31 = graycoprops(glcms3,'contrast');
feature(11)=c31.Contrast;
c41 = graycoprops(glcms4,'contrast');
feature(12)=c41.Contrast;

%calculation of energy

c51 = graycoprops(glcms1,'energy');
feature(13)=c51.Energy;
c61 = graycoprops(glcms2,'energy');
feature(14)=c61.Energy;
c71 = graycoprops(glcms3,'energy');
feature(15)=c71.Energy;
c81 = graycoprops(glcms4,'energy');
feature(16)=c81.Energy;
combined_features=sum(feature);

%Normalization of features extracted
for l=1:16
    average(l)=(feature(l)/combined_features);
end
mean=mean(average);
SD=std(average);
for r=1:16
    Output_data=(average-mean)/SD;
end
fid =fopen('tester2.txt','w');
for i=1:16
    fprintf(fid,'%d:%f	' ,i,Output_data(i));
end
fclose(fid);
B.2 Source to determine morphological parameters of PM image.

close all;
clear all;
clc;
J = imread('app_G1_start_010.jpg');
%median filter to eliminate noise
I1 = medfilt2(J,[3 3]);
I2 = medfilt2(I1,[3 3]);
I3 = medfilt2(I2,[3 3]);
I4 = medfilt2(I3,[3 3]);
I5 = medfilt2(I4,[3 3]);
I = medfilt2(I5,[3 3]);
%Sobel is selected as the optimal segmentation algorithm to further measure the

%physical properties.
%step1:Binary gradient mask
[junk threshold] = edge(I, 'sobel');
BWb = edge(I,'sobel', threshold);

%step2:Dilated gradient mask
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 0);
BWsdil = imdilate(BWb, [se90 se0]);

%Step3:Binary image filled with holes
BWdfill = imfill(BWsdil, 'holes');
figure, imshow(BWdfill);
title('binary image with filled holes');

%Step4:Final segmented image
seD = strel('diamond',1);
BWsegmented = imerode(BWdfill,seD);
BWsegmented = imerode(BWsegmented,seD);
BWfinal = bwareaopen(BWsegmented,50);
[L, NUM] = bwlabel(BWfinal,8);

% pixel to meter conversion
pix2um = 1280/400;

%Area measurement
parameter1 = regionprops(L,'Area');
area = cat(1, parameter1.Area);
%Perimeter measurement
parameter2=regionprops(L,'Perimeter');
perimeter=cat(1,parameter2.Perimeter);
%Major axis measurement
parameter3=regionprops(L,'MajorAxisLength');
major_axis=cat(1,parameter3.MajorAxisLength);
%Minor axis measurement
parameter4=regionprops(L,'MinorAxisLength');
minor_axis=cat(1,parameter4.MinorAxisLength);
%Conversion of pixels into micrometer  
Area_value=pix2um*area;  
Perimeter_value=pix2um*perimeter;  
Major_value=pix2um*major_axis;  
Minor_value=pix2um*minor_axis;  

%Measurement of aspect ratio and shape  
for z=1:NUM  
Aspect_ratio(z)=Major_value(z)/Minor_value(z);  
Form_factor(z)=(4*pi*Area_value(z))/(Perimeter_value(z)^2);  
end

82
Appendix –C

Haralick features

Angular Second Moment:

\[ f_1 = \sum\sum (p(i,j))^2 \]

Contrast:

\[ f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\} \]

Correlation:

\[ f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \]

Sum of Squares- Variances:

\[ f_4 = \sum\sum (i - \mu)^2 p(i,j) \]

Inverse difference moment:

\[ f_5 = \sum\sum \frac{1}{1 + (i - j)^2} p(i,j) \]

Sum average:
\[ e = \sum_{i=2}^{2N_g} ip_{x+y}(i) \]

Sum Variance:

\[ f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i) \]

Sum Entropy:

\[ f_8 = -\left\{ \sum_{i=2}^{2N_g} p_{x+y}(i) \log \{ p_{x+y}(i) \} \right\} \]

Entropy:

\[ f_9 = -\left\{ \sum_i \sum_j p(i,j) \log \{ p(i,j) \} \right\} \]

Difference Variance:

\[ f_{10} = -\left\{ \sum_{i=0}^{N_g-1} (i - \mu_{x-y})^2 p_{x-y}(i) \right\} \]

Difference Entropy:

\[ f_{11} = -\left\{ \sum_{i=0}^{N_g-1} p_{x-y}(i) \log \{ p_{x-y}(i) \} \right\} \]

Information Measures of correlation:

\[ f_{12} = \frac{HXY - HXY1}{\max \{HX, HY\}} \]

\[ f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{\frac{1}{2}} \]

\[ HXY = -\left\{ \sum_i \sum_j p(i,j) \log \{ p(i,j) \} \right\} \]
where $H_X$ and $H_Y$ are entropies of $p_x$ and $p_y$, and

$$H_{XY1} = -\left\{ \sum_i \sum_j p(i, j) \log \{p_x(i) p_y(j)\} \right\}$$

$$H_{XY2} = -\left\{ \sum_i \sum_j p_x(i)p_y(j) \log \{p_x(i)p_y(j)\} \right\}$$

Maximal correlation coefficient:

$$f_{14} = \left( \text{Second largest eigen value of } Q \right)^{\frac{1}{2}}$$

where

$$Q(i, j) = \sum_k \frac{p(i, k) p(j, k)}{p_x(i)p_y(k)}$$
Appendix –D

Image classification using histogram based SVM and GLCM based SVM

Figure D-1: Schematic diagram showing the image classification by SVM classifier trained by the features extracted using the first-order histogram and second-order GLCM.

86