Predicting the Particle Size Distributions of Spherical Particle Sets from Synthetic Images: A Comparison of 9 Classic Image Features

Javad Manashti[1,*], François Duhaime[1], Matthew F. Toews[2], Pouyan Pirnia[1]

[1] Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure

[2] Systems Engineering Department, École de technologie supérieure, Montréal, Canada

Abstract

The two objectives of this paper were to demonstrate use the of the discrete element method for generating synthetic images of spherical particle configurations, and to compare the performance of 9 classic feature extraction methods for predicting the particle size distributions (PSD) from these images. The discrete element code YADE was used to generate synthetic images of granular materials to build the dataset. Nine feature extraction methods were compared: Haralick features, Histograms of Oriented Gradients, Entropy, Local Binary Patterns, Local Configuration Pattern, Complete Local Binary Patterns, the Fast Fourier transform, Gabor filters, and Discrete Haar Wavelets. The feature extraction methods were used to generate the inputs of neural networks to predict the PSD. The results show that feature extraction methods can predict the percentage passing with a root-mean-square error (RMSE) on the percentage passing as low as 1.7%. CLBP showed the best result for all particle sizes with a RMSE of 3.8 %. Better RMSE were obtained for the finest sieve (2.1%) compared to coarsest sieve (5.2%).

Keywords: image analysis, feature extraction, particle size distribution, discrete elements, neural network

[*] Corresponding author: Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, 1100 Notre-Dame Ouest, Montreal, Quebec, H3C 1K3, Canada.
Email: mjmanashti@gmail.com

1 Introduction

Sieving, the current standard method for PSD determination in geotechnical engineering [1], dates back to the first half of the 20th century. With this method, the specimen is poured in a series of sieves and shaken for a predetermined duration [2]. The cumulative mass passing each sieve is presented against the sieve opening on a logarithmic scale. Several useful parameters can be extracted from the PSD, such as the characteristic sizes corresponding to specific percentages passing (e.g. D_{50}, the particle size for which 50 % of the granular material mass is smaller), and parameters describing the shape of the PSD (e.g. $C_u = D_{60}/D_{10}$, the coefficient of uniformity). These parameters are used for soil classification and the prediction of soil properties (e.g. [3]). Sieving yields accurate results, but is time-consuming.

Several methods have been developed to determine the PSD based on soil photographs and image analysis techniques. These methods can be classified in two main groups. Methods of the first group use image segmentation to define the contour of each particle in the photograph. The size of each segmented zone is then used to estimate the PSD. Methods of the second group extract textural features from soil photographs to quantify the PSD [4].

Commercial codes such as WipFrag [5] and VisioRock [6] use image segmentation to determine the particle size distribution of granular materials from photographs. The relationship between the size of the segmented zone in the photographs and the real particle size is not straightforward as segregation, capturing and overlapping errors will cause the segmented zone and particle size distributions to differ [7, 8]. Segregation refers to the reorganization of particles according to their size when they are displaced, for instance when they are pushed down a slope by machinery or when they are shaken. Capturing refers to the unequal probability of particles of different sizes of being observed on a surface. For example, a particle that has a diameter of the same order as the sample or pile is much more likely to be visible at the surface compared to a small particle. Overlapping errors are due to particle superposition. Particles are hiding each other. Even if some methods are available to take these errors into account, they remain largely unresolved [8].

The accuracy of segmentation methods differs depending on the parameters that are compared, the reference method, and the granular material used in the comparison. The parameter that is most often compared is the D_{50}. For 10 digital images of blasted sandstone, Sudhakar et al. [9] compared automatic segmentation using commercial codes WipFrag and Fragalyst [10], with manual segmentation on tracing paper. They did not compare their PSD with sieving results. Their comparison focused on segmentation errors and avoided the influence of segregation, capturing and overlapping errors. The difference between the D_{50} of WipFrag, Fragalyst, and the manual segmentation was below 10 % for half of the digital images. Errors of up to a factor 2 were observed on some of the other images. Liu and Tran [11] compared the PSD obtained with commercial segmentation codes FragScan [12], WipFrag [5] and Split [13], with sieving results for dry waste rock from a mining operation. The ratio between the D_{50} for the three commercial codes and the reference D_{50} obtained by sieving varied between 1.3 and 3.2. On average, Split and WipFrag overestimated the D_{50} by around 55%, while the overestimation by FragScan was around 100%. The PSD given by Sudhakar et al. [9] and Liu and Tran [11] also allow the root mean square error (RMSE) on the percentage passing for each size class to be estimated and compared. RMSE values between 13 and 36 % were obtained when comparing FragScan, WipFrag and Split for dry waste rock [11]. When comparing only the segmentation, RMSE of 15 and 20 % were obtained for WipFrag and Fragalyst [9].

Textural features are functions that describe the spatial arrangement of pixel intensities [14, 4]. Methods for the extraction of textural features can be divided in three main categories [15]. Methods of the first type use statistical parameters to describe the spatial distribution of pixel intensities. For example, the Haralick features [16] use statistics (e.g. average, variance, correlation) of the global co-occurrence matrix that describes the number of pixels of given intensities separated by a given distance. Methods of the second type are based on local patterns found in the image (e.g. local binary pattern (LBP) [17]). The third type includes transform-based feature extraction methods, such as Gabor wavelets [18].

Statistical textural parameters have been used for PSD determination or rock classification. Itoh et al. [19] compared the performance of 32 textural and colour features, including Haralick features and run-length statistics, for the prediction of particle size. Five materials with different uniform PSD were

photographed under 22 different lighting levels for a total of 3960 images. An analysis of variance showed that color properties, like saturation and hue, are sensitive to changing illumination. Haralick features and run-length statistics applied on edge-enhanced images were found to be better predictor of the particle size.

Local pattern methods describe the relationship between the intensity of the gray levels in the neighborhood of each pixel of an image. The Local Binary Pattern (LBP) method is a good example [17]. With this method, the relative intensity of each pixel in the neighborhood of a center pixel is encoded as a binary number. Statistics are then computed on the occurrence of each binary pattern in the image. Local pattern methods can be rotation invariant and independent of illumination [17]. These methods are generally used for classification purposes. For example, Paclík et al. [20] used LBP, among other methods, in a rock classification system. To our best knowledge, local pattern methods have never been used to determine the PSD of granular materials.

Transform-based methods allow features to be extracted from images in the frequency domain. The most common method for PSD determination is the Harr wavelet transform [21, 22]. Wavelet transforms give local information on the frequency content of an image. The wavelet transformation of an image involves multiplying the local pixel intensities in an image by a mother wavelet. The parts of the image where the frequency content corresponds to the frequency of the wavelet will show a stronger response. Shin and Hryciw [22] compared the accuracy of two-dimensional Haar wavelet and Haralick features for PSD determination. Both wavelet transforms and Haralick features were dependent on the illumination level and particle colour. However, the normalization of the wavelet energy was shown to eliminate the effect of illumination and particle color. Hryciw et al. [21] also presented the theoretical basis behind the use of wavelet analyses for PSD determinations. Yaghoobi et al. [23] determined the particle size of fragmented rock using features extracted from Fourier transforms, wavelet transforms, Gabor filter and their combinations. With Fourier transforms, the mean and standard deviation of the magnitude spectrum for 20 rings of equal thickness was used for each image. The best performances were obtained with Fourier transforms, followed by Gabor filters, and a combination of Fourier and wavelet transforms. Size estimation was better for fine to medium particles.

Several researchers have combined image processing methods with Artificial Neural Networks (ANN) for PSD determination [24, 25, 26]. Because of the structure of classic ANN, it is generally not possible to use the intensity of each pixel directly as ANN inputs. Consequently, preprocessing techniques have been used to extract parameters, such as textural parameters, from the images to be used as ANN inputs. Ghalib et al. [25] used a selection of eight Haralick features as the input for an ANN to predict the average size of sub-angular sand particles. Their dataset included 220 images. The particle size RMSE varied between 1.4 and 6.1 %. In another study, Ko and Shang [27] defined a uniformity coefficient that was combined with an initial PSD estimate obtained with commercial code WipFrag as inputs for an ANN. The ANN output was the particle sizes corresponding to weight percentages passing of 50, 75, and 90 % (D_{50}, D_{75} and D_{90}). The uniformity coefficient was obtained through the thresholding of 5 areas in each photograph (center, top left, right and bottom). The uniformity coefficient was defined as the difference in the proportion of darker pixels in the center area with respect to the four surrounding areas. Only 9 images were used for the ANN training. The ratio between the D_{50}, D_{75} and D_{90} estimated from the neural network and the target values obtained from sieve analyses varied between 0.7 and 1.4. Hamzeloo et al. [28] used image processing in combination with an ANN to predict the PSD of crushed rocks on a conveyor belt. They segmented their images manually after several image processing steps (e.g. image sharpening, edge detection, thresholding). They then calculated different geometrical features for the segmented particles (e.g. Ferret diameter). A Principal Component Analysis (PCA) was then used to select the best features and to use them as the input to a series of ANN. They compared their results with sieve analyses. Their dataset included 21 sieve analyses. Yaghoobi et al. [23] used Fourier transforms, Gabor filters, wavelet transforms, and their combinations to extract image features that were later used as inputs for neural networks to determine the PSD of fragmented rock. Their database included 226 photographs. Manual segmentation results obtained with commercial code Split were used as the target for the neural network.

The training of neural networks requires large datasets that are not readily available. The previously cited examples are based on a limited number of images, generally much less than 1000, and they involve a limited number of input parameters. Because they involve different granular materials and

different targets for the neural network (i.e. sieving results or manual segmentation results), it is also difficult to compare their results.

The main objectives of this paper are 1) to compare the accuracy of various textural parameters in predicting the PSD of materials composed of spherical particles and 2) to introduce a new dataset of synthetic images of granular materials generated using the discrete element method. The textural parameters are used as the input for a series of neural networks. The selected parameters include some that have never been tested for PSD analyses, notably LBP and the related methods, Entropy, and Histograms of Oriented Gradients (HOG). These new methods are compared with more established methods like Haralick parameters, Haar wavelet transforms, Fourier transforms, and Gabor filters. The database presented in this paper contains 53 003 pairs of images (top and bottom viewpoints) of spherical particle configurations created with the DEM.

The paper first describes the synthetic dataset, the feature extraction methods and the neural network parameters. The paper then presents the results and compares the accuracy of each method with previously published results. The synthetic dataset is shown to facilitate the comparison of a different image analysis methods.

2 Methodology

2.1 Synthetic granular image dataset

A large image dataset was prepared by Pirnia et al. [29] to compare the feature extraction methods and for training, validation, and testing of the neural networks. The discrete element (DE) code YADE was used to generate a large number of synthetic photographs of granular materials. A text file containing a list of 53003 PSD with particle sizes between 75 and 1180 μm was created. The list was obtained by considering all possible weight percentage combinations with 5 % increments for 5 sieves (106, 150, 250, 425 and 710 μm). COMSOL's Java interface was used to read the data and control YADE's python interface to generate the DE specimen [30]. The particles were created in a transparent virtual box. They were given random colors. The particles were allowed to settle in the box. Pictures were taken using two virtual cameras from the top and bottom of the box (Fig. **Error! Reference source not found.**).

For this paper, all pictures were converted to grayscale images. A total of 53003 pairs of top and bottom images sized 128 x 128 pixels were produced. Image pairs were merged into 256 x 128-pixel images, which were used as the input of the model. Because each particle has a finite mass, the real PSD of each image can differ slightly from the specified PSD. However, the RMSE of the percentage passing for each sieve was found to be smaller than 0.2 %.

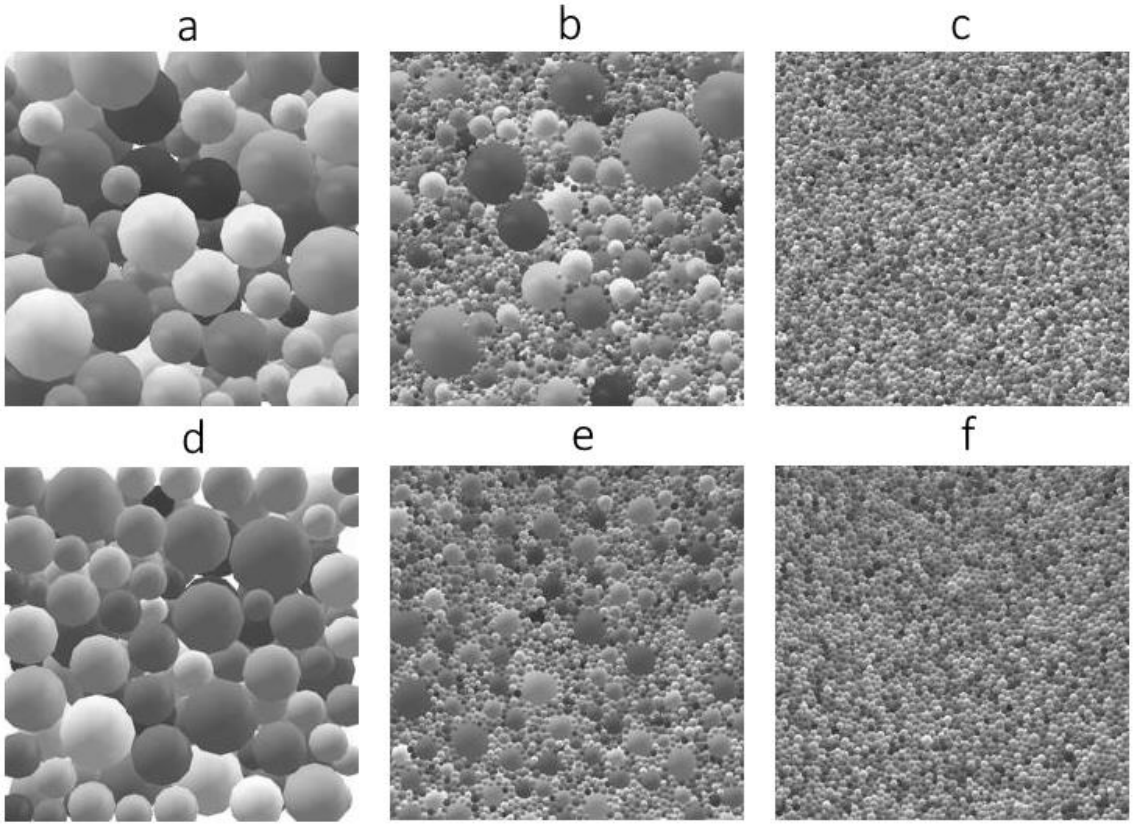

Fig. 1 Top (upper row, a-c) and bottom (lower row, d-f) views of the synthetic particles in the virtual transparent box. Left (a, d), middle (b, e) and right (c, f) images show examples of coarse, widely graded, and fine PSD

2.2 Feature Extraction methods

The following sections describe the feature extraction methods that were applied on the dataset. These feature extraction methods were used to generate features that were fed to a series of artificial neural networks presented later in the paper.

2.2.1 Haralick Features

Haralick features are statistics of the gray level cooccurrence matrix (GLCM) [31, 16, 19]. Item (*i, j*) in the GLCM gives the number of pixels with intensity level *i* that have a specified spatial relationship with other pixels with intensity level *j*. The GLCM has two important parameters: offset and number of gray levels (N_g). The offset defines the spatial relationship of the two pixels. Horizontal offsets of 1 and 10 pixels ([0 1] and [0 10]) were tested in this study. The number of gray levels determines the size of the GLCM. GLCM were calculated for both 8 and 64 gray levels.

The 14 textural features defined by Haralick et al. [16] were calculated for each GLCM (total of 56 features) using the MATLAB script of Monzel [32]. Preliminary tests showed that some Haralick features are better correlated with the percentage passing. This is the case of correlation and contrast:

$$Correlation = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} i\ j\ GLCM(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y} \quad (1)$$

$$Contrast = \sum_{n=0}^{N_g-1} n^2 \sum_{\substack{j=1}}^{N_g}\sum_{\substack{i=1 \\ |i-j|=n}}^{N_g} GLCM(i,j) \quad (2)$$

where μ_x and σ_x are respectively the mean and standard deviation of $p_x(i)$, and μ_y and σ_y are the mean and standard deviation of $p_y(j)$, respectively:

$$\boldsymbol{p_x(i) = \sum_{j=1}^{N_g} GLCM(i,j)} \quad (3)$$

$$\mathbf{p_y(j) = \sum_{i=1}^{N_g} GLCM(i,j)} \quad (4)$$

Correlation describes the linear dependency in pixel intensity and contrast increases when the image contains abrupt changes in pixel intensity [31].

2.2.2 Histogram of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) describe the image by counting the occurrence of different orientations of the gray intensity gradient for a series of predefined cells [33]. The cells can be square or rectangular. They are defined by their size. The gradients are assigned to bins. The gradient magnitude is used to weight the gradient when they are assigned to the bins: higher gradients have a higher weight. Larger blocks that include several cells are also defined to normalize the content of each bin by the norm of all bins in the block. Normalizing allows local contrast variations in an image to be taken into account. Cells that are smaller, or similar in size to the particles will tend to be dominated by

a small number of particle edges that will give a preferential orientation to the gradient. Cells that are larger than the particle size will result in a more random gradient orientation [34]. Fig. 2 shows a comparison of the HOG presented as rose diagrams for the coarsest PSD and a finer PSD for a cell size of 8 pixels. The gradient orientation is more random for the fine PSD.

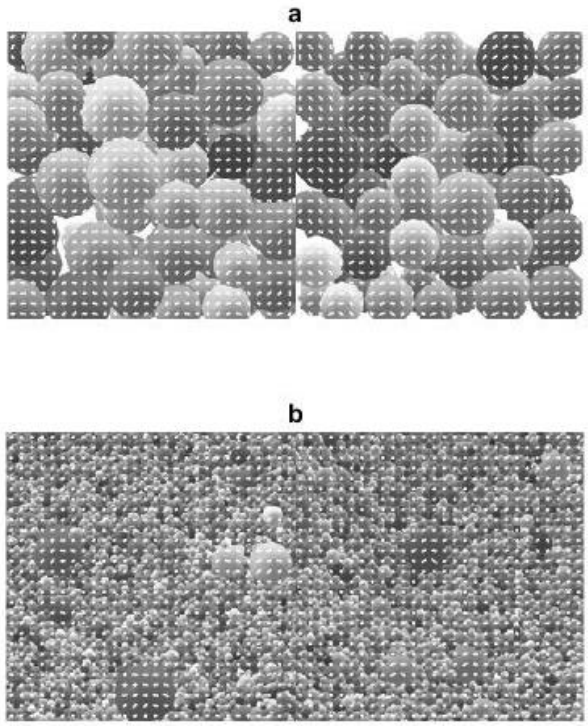

Fig. 2 Examples of gradient orientations calculated for a cell size of 8 pixels for a) coarse and b) fine PSD

In this study, square HOG cells with sizes of 2, 4, 8, 16, 32, and 64 pixels were used. MATLAB function *extractHOGFeatures* [35] was used to generate the HOG features. The mean and standard deviation of all HOG for each size were calculated to produce 12 features that were used as the ANN input. The normalizing blocks were defined to include 4 cells with an overlap of one cell width for contiguous blocks. Nine orientations bins were used.

2.2.3 Local entropy of grayscale image

Entropy [36] is a measurement of the randomness of the gray level intensity in the neighborhood of a pixel. Entropy is a statistical parameter related to the image texture. The entropy of a grayscale image can be calculated as follows:

$$H(X) = -\sum_{j=1}^{N_g} P_j \ log_2 P_j \quad (5)$$

where $H(X)$ is the entropy of the neighborhood X around a center pixel, N_g is the number of gray level bins, and P_j is the probability of occurrence of gray level j in the neighborhood X [37].

The neighborhood around the center pixel is defined using a filter. Fig. 3 shows the filters that were used in this study. These filters were defined using the MATLAB functions *getnhood* and *strel*. The filter diameters vary between 1 and 9 pixels. The entropy was calculated with MATLAB function *entropyfilt* with bin numbers varying between 2 and 256. **Error! Reference source not found.** presents the effect of the disk filter size and number of bins on $H(X)$. Coarser particles correspond to lower values of the local entropy. The mean and standard deviation of $H(x)$ were calculated for the 9 disk filters (total of 18 features) and used as the ANN input.

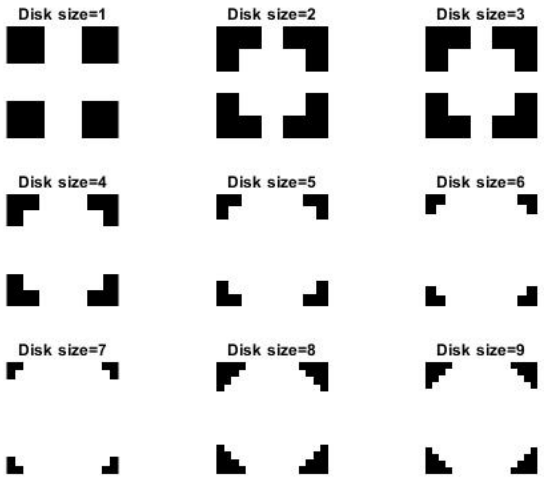

Fig. 3 Nine disk filter sizes used to calculate entropy

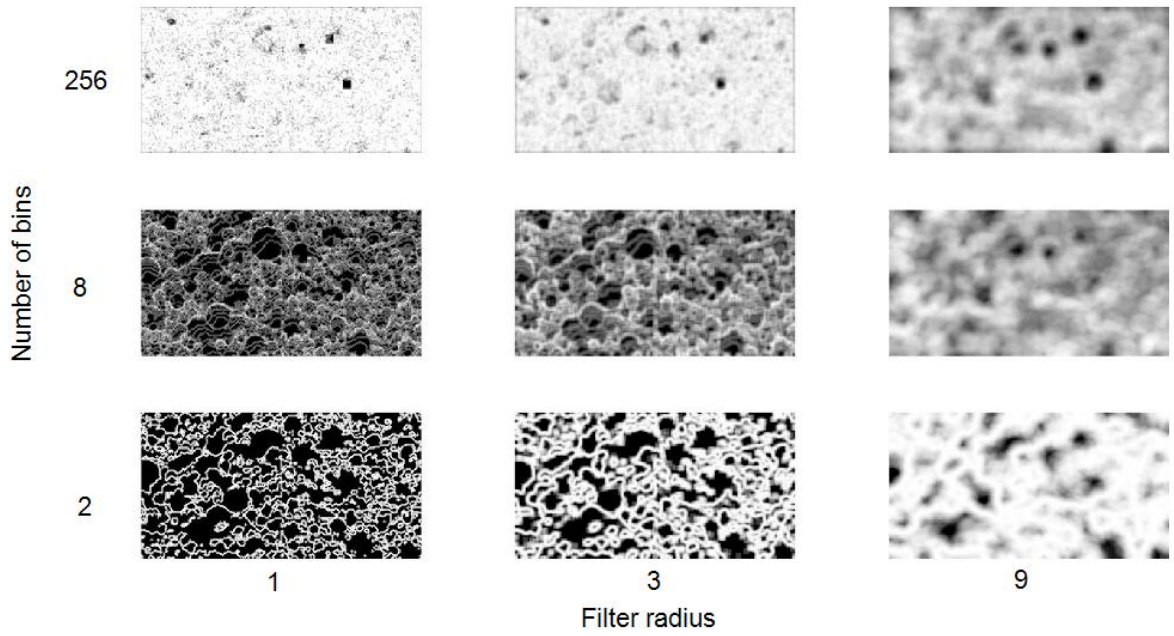

Fig. 4 Effect of disk filter size and number of bins on the local entropy for the widely graded PSD (**Error! Reference source not found.**b and 1e). Coarser particles are associated with lower values of the local entropy (darker gray levels)

2.2.4 Local Binary Pattern (LBP)

The Local Binary Pattern (LBP) sets the center pixel of a neighborhood as a threshold [17]. Neighboring pixels take a value of 0 or 1 based on the gray level difference between them and the center pixel [38]. This pattern is encoded into a binary value by multiplying the 0 or 1 values by 2^p, where p is the neighboring pixel number. For example, the LBP for 8 neighboring pixels corresponds to an 8-bit number. LBP is calculated for each pixel of an image with this relationship:

$$LBP = \sum_{p=0}^{p-1} s(x_p - x_c)\, 2^p \tag{6}$$

where x_c is the center pixel gray level and x_p are the gray levels for the neighboring pixels. $s(x_p - x_c)$ is the step function:

$$s(x_p - x_c) = \begin{cases} 1, x_p - x_c \geq 0 \\ 0, x_p - x_c < 0 \end{cases} \tag{7}$$

In its original form, LBP uses the closest 8 pixels around the center pixel. In later LBP implementations, different numbers of neighboring pixels and distances between the center and neighboring pixels can be chosen [39]. To achieve rotation invariance, the encodings that only differ by a rotation can be combined into bins, thus limiting the number of features. In this project, the LBP features were calculated for radius values of 1, 2, 3, 4, 5, 6, 8, and 12 pixels, and for 4, 6, 8, 12, and 16 neighbors to find the best filters for the fine and coarse particles.

2.2.5 Local configuration pattern (LCP)

The Local Configuration Pattern (LCP) combines LBP with microscopic configurations (MiC) [40, 41]. LBP encodes the structure of the neighborhood around a center pixel based on a step function that does not include any information on the variability of the gray level intensity. MiC adds information on this variability. With MiC, the value of the center pixels for each rotation invariant LBP bin is defined as a linear combination of the neighborhood pixel intensities multiplied by a set of weights:

$$\begin{bmatrix} x_{c,0} \\ \vdots \\ x_{c,N} \end{bmatrix} = \begin{bmatrix} x_{0,0} & \cdots & x_{p,0} \\ \vdots & \ddots & \vdots \\ x_{p,N} & \cdots & x_{p,N} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_p \end{bmatrix} \quad (8)$$

where $x_{c,0}$ through $x_{c,N}$ are the center pixels in the current LBP bin, $x_{0,0}$ through $x_{p,N}$ are the neighborhood pixels and a_0 through a_p are the weight for the neighborhood pixels. The weight are determined using a least-square method. A Fourier transform is used to make the weights rotation invariant. The LCP features combines the LBP histogram with the MiC weights. In this project, LCP features were calculated with radius of 1, 2, or 3 pixels and neighborhoods of 1, 2, 3, 4, 5, 6, 7, 8, or 12 pixels.

2.2.6 Completed Local Binary Pattern (CLBP)

The Completed Local Binary Pattern (CLBP) has three outputs: CLBP-Center, CLBP-Sign, and CLBP-Magnitude. CLBP-Sign corresponds to LBP [42]. CLBP-Center describes the gray level of the center pixel. It is converted into a binary value via global thresholding using the mean gray level for the image. CLBP-magnitude is calculated using the absolute value of the difference between the gray

level of the neighboring pixels and the center pixel ($|x_p - x_c|$). The absolute difference for each neighboring pixel is converted to a binary value by thresholding using the mean absolute difference for the image. These binary values are encoded in a number similarly to LBP. In this project, the radius was varied between 1 and 3. The number of neighbors ranged from 1 to 6.

The MATLAB codes used for the calculation of LBP [17], LCP [41], and CLBP [43] were obtained from the Center for Machine Vision Research, Department of Computer Science and Engineering, University of Oulu, Finland [44].

2.2.7 Fourier Transform

The two-dimensional discrete Fourier transform $F(u, v)$ of an image with pixel intensity f(x, y) is defined as follows [45, 23]:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{n-1} f(x,y) e^{-2\pi\sqrt{-1}\left(\frac{ux}{M}+\frac{vy}{N}\right)} \quad (9)$$

where M×N is the image size, u and x=0, 1, 2, …, M-1 and v and y=0, 1, 2, …, N-1. The magnitude and standard deviation of the 2D FFT was used to find the PSD.

Error! Reference source not found. shows the magnitude spectrum of the Fourier Transform for the images shown in Figs. 1a, 1b and 1c. The center of the magnitude spectrum is associated with the coarse particles while the surrounding area is associated with the fine particles. The brighter center of the magnitude spectrum in Fig. **Error! Reference source not found.**a is associated with coarser particles in the original image. This brighter center can be compared with the darker center in Fig. **Error! Reference source not found.**c associated with fine particles.

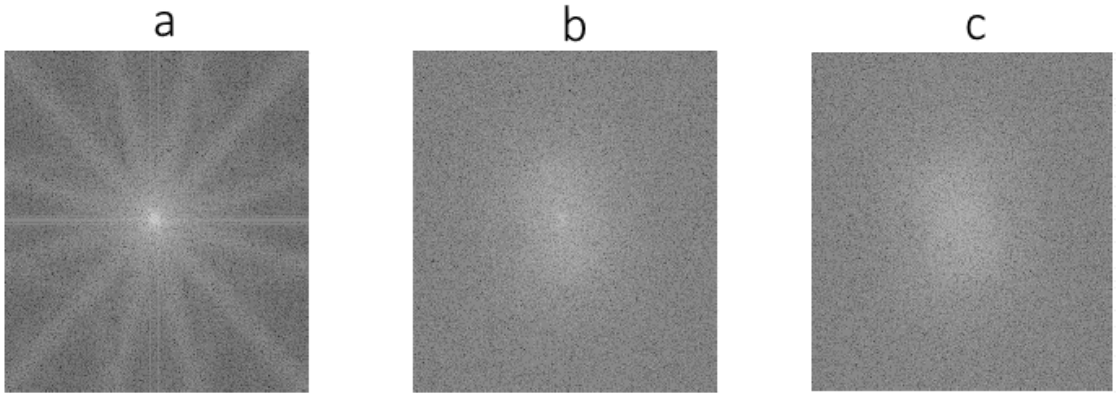

Fig. 5 Magnitude spectrum of the Fourier transform for images corresponding to the coarse (a), widely graded (b), and fine (c) PSD in Figures 1a, 1b, and 1c, respectively

The particle size distribution can be related to the mean value and standard deviation of the Fourier transform magnitude for a set of ring filters [23]. In this study, the mean value and the standard deviation of the magnitude were calculated for different sets of ring filters. Two parametric studies were conducted. In the first, the number of rings was set to five and the thickness of consecutive rings was multiplied by a constant increment X between 0.8 and 2.4. Two examples are shown in Fig. 6 for X=1.0 (first row) and X=2.0 (second row). In the second method, the thickness of each ring was constant, and the number of ring filters was varied between 5 and 20. The third row in Fig. 6 shows eight rings with a constant thickness.

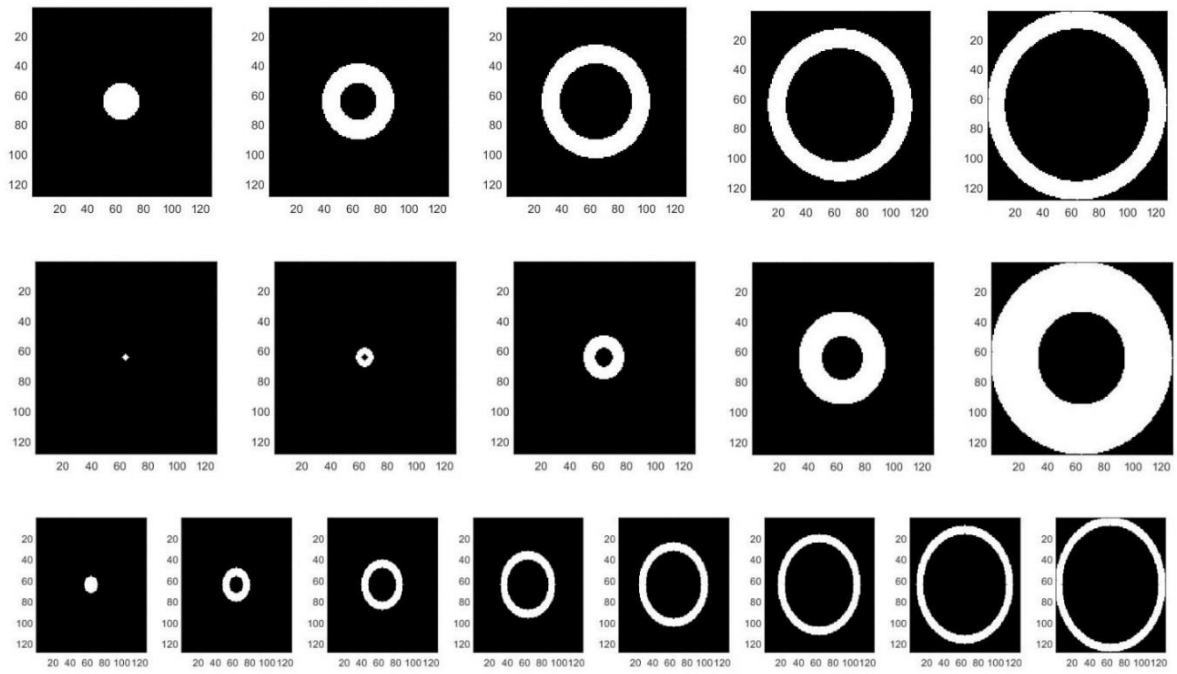

Fig. 6 Examples of concentric circular filters for which the mean and standard deviation of the Fourier transform magnitude spectrum is calculated. The first row shows five filters with equal thickness (X=1.0). The second row illustrates five filters in which the thickness of every ring is twice the previous ring thickness (X=2.0). The third row shows eight filters with an equal thickness

The mean value and standard deviation of the magnitude were used as the input for an ANN. Average values were calculated for the top and bottom images using the same ring filters. The total number of inputs for the ANN is equal to twice the number of rings. It varies between 10 and 40 for the two parametric studies.

2.2.8 Gabor Filters

Gabor filters with different wavelengths and orientations were applied to the images to extract features to determine the PSD. Gabor filters combine a Gaussian kernel function with a sinusoidal plane wave [4, 23]. They allow the local frequency content of an image to be evaluated. A two-dimensional Gabor function ($F(x, y)$) and its two-dimensional Fourier transform ($F(u, v)$) can be defined as follows:

$$F(x,y) = \left(\frac{1}{2\pi S_x S_y}\right) exp\left(-\frac{1}{2}\left(\frac{X^2}{S_x{}^2} + \frac{Y^2}{S_y{}^2}\right)\right) cos(2\pi u_0 X) \tag{10}$$

$$F(u,v) = \frac{1}{2}\left\{exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)}{S_u^2} + \frac{v^2}{S_v^2}\right]\right\} + exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)}{S_u^2} + \frac{v^2}{S_v^2}\right]\right\}\right\} \tag{11}$$

$$S_u = \frac{1}{2\pi S_x}, S_v = \frac{1}{2\pi S_y} \tag{12}$$

where S_x and S_y are the Gaussian function standard deviations in the *x* and *y* directions, respectively. S_u and S_v are the Gaussian function standard deviation in the *u* and *v* directions, respectively, and u_0 is the sinusoidal wavelet central frequency in the *x* direction.

Fig. 7 shows the Gabor filter magnitude for the wavelength of 3,9, and 15 pixels and the orientation of 0°, 90°, and 150°, for the top images corresponding to the coarse (top row) and fine (second row) PSD. A filter bank corresponding to different wavelengths (3 to 15 pixels) and orientations (0 to 150°) was prepared using MATLAB function Gabor. The mean and standard deviation of a set of 42 filters was used as the input of the ANN (84 parameters). MATLAB function imgaborfilt was used to apply the filter bank to the images.

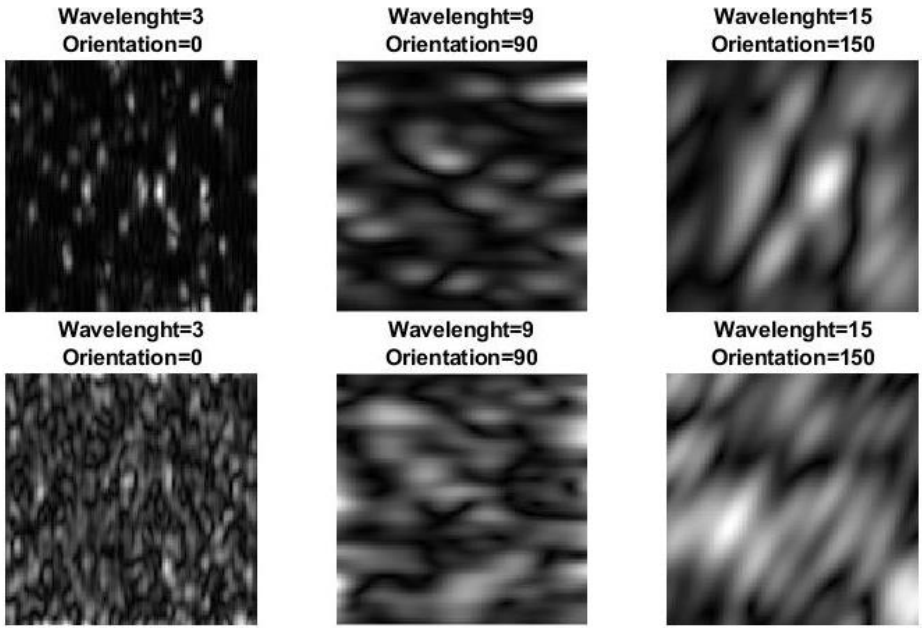

Fig. 7 Magnitude of 3 selected Gabor filters with different wavelengths and orientations for the coarse (top row) and fine (second row) PSD

2.2.9 Haar wavelet transforms

Wavelet transforms use mother wavelets (Ψ) for the decomposition of a signal. Mother wavelets are functions that are localized in space and in the frequency domain. The Haar mother wavelet can be described as follows [36]:

$$\psi_{u,v} = \begin{cases} \sqrt{u} & 0 \le ux - v \le 0.5 \\ -\sqrt{u} & 0.5 \le ux - v \le 1.0 \\ 0 & elsewhere \end{cases} \quad (13)$$

where *u* is a scaling factor and *v* is a translation of the mother wavelet on the x axis. Detailed calculations of 2D Haar wavelet transforms of images in the context of PSD determinations are presented by Hryciw et al. (2015). Transforms can be applied successively to 4×4 pixel regions (*I*) that are defined as follows [36]:

$$I = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (14)$$

Applying the transform results in the following four coefficients:

$$A = \frac{a+b+c+d}{2} \quad (15)$$

$$H = \frac{a+b-c-d}{2} \quad (16)$$

$$V = \frac{a-b+c-d}{2} \quad (17)$$

$$D = \frac{-a+b+c-d}{2} \quad (18)$$

where *A*, *H*, *V* and *D* are respectively the approximation, horizontal, vertical and diagonal coefficients. Each application of the wavelet transform decreases the resolution of the image by a factor 2. Transforms can thus be applied 7 times for 128×256 pixel images. At each new level, the transform is applied to the previous approximation matrix.

The square of each coefficient can be used to describe the energy of the image. The sum of the energy associated with the *A*, *H*, *V* and *D* matrices is preserved after each transform. Consequently, each new level results in three independent energy values [46]. For each level, the mean and standard deviation of the energy associated with the *H*, *V* and *D* coefficients were used as features. This results in 49 features for the 7 levels. MATLAB function haart2 was used to compute the coefficients. **Error!**

Reference source not found. presents one example of the successive application of Haar wavelet transforms and the resulting approximations.

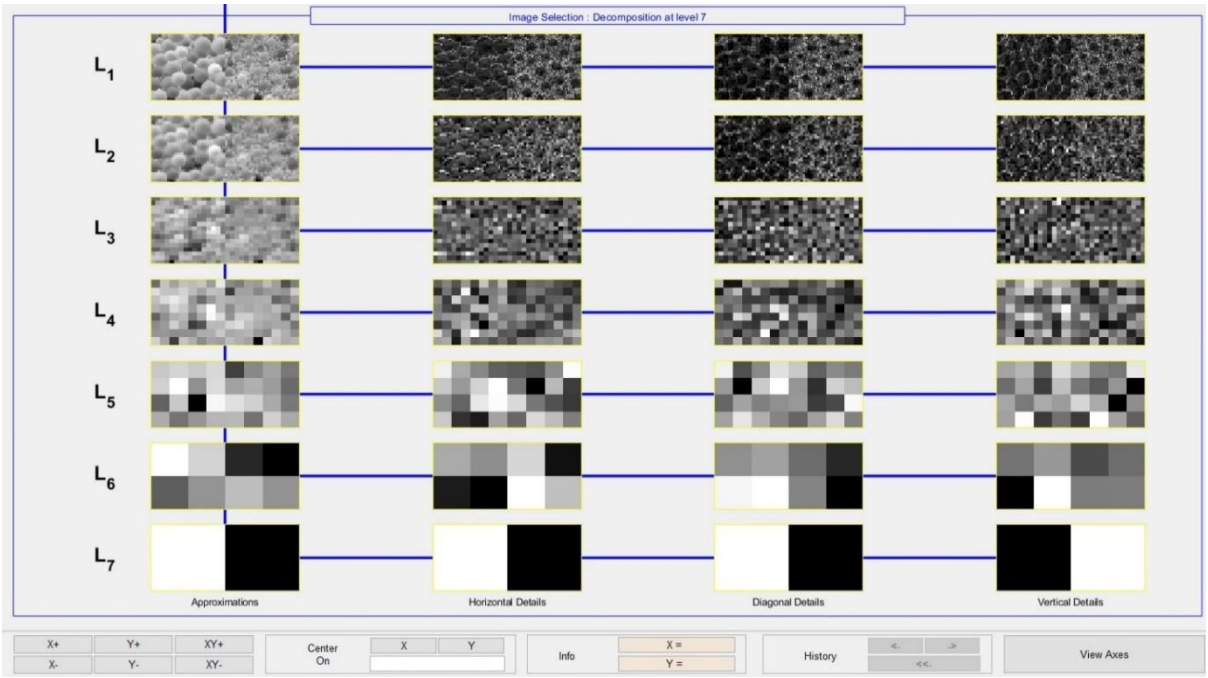

Fig. 8 Seven levels of wavelet approximations for horizontal, diagonal, and vertical orientations

2.3 Artificial Neural Networks

The features that were extracted for the 53 003 images were used as the input of a series of neural networks to predict the percentage passing for sieves of 106, 150, 250, 425 and 710 µm (Fig. **Error! Reference source not found.**). Function fitting neural networks were trained with MATLAB to predict the particle size based on the extracted features. For all methods in this study, one hidden layer with 10 neurons was used. A hyperbolic tangent sigmoid was used as the transfer function.

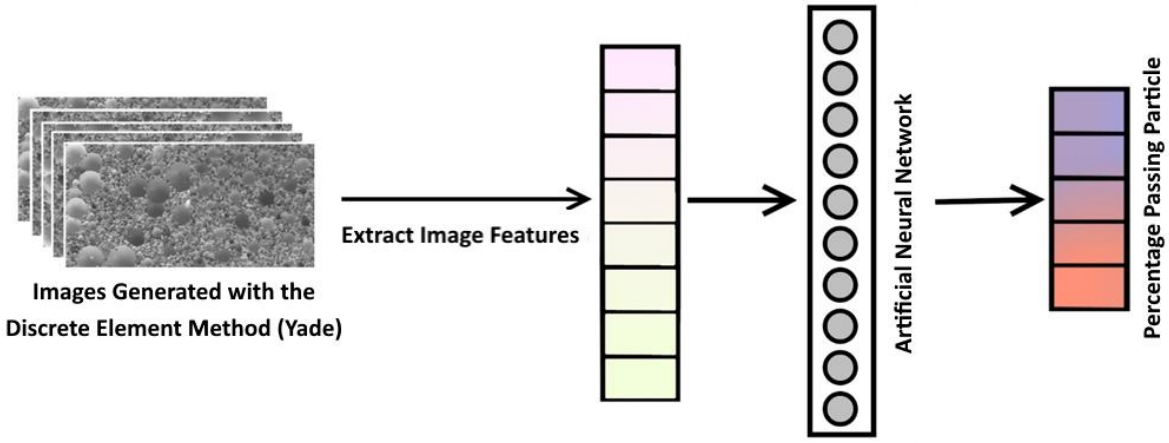

Fig. 9 Artificial Neural Network to predict the percentage passing for a series of sieves

For optimal training of the neural networks, 70% of the 53 003 images were used for training, 15% for validation, and 15% for testing. The root mean square error (RMSE) on the percentage passing was used to calculate the error associated with the neural network. The same parameter is used in the presentation of the results to compare the performance of each type of features. The Scaled Conjugate Gradient method was used for the network training. **Error! Reference source not found.** illustrates the performance of the neural network during training for a selection of the 618 best features for all methods. The RMSE decreases for the training, test, and validation datasets during the network training until the maximum number of epochs is reached (5000).

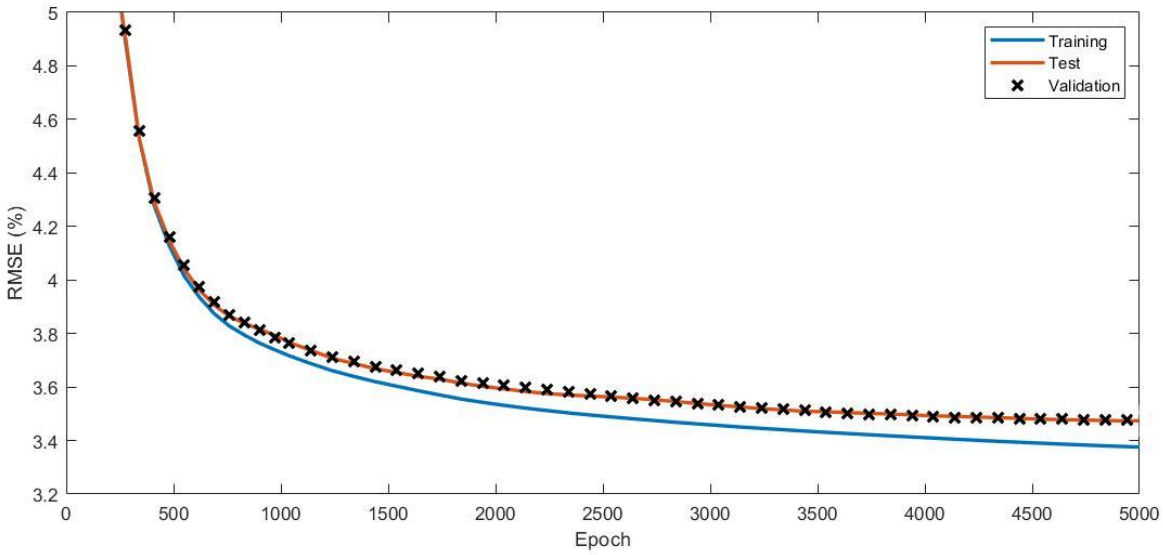

Fig. 10 Neural network training performance for a selection of the 618 best features

Larger networks were also tested for the wavelet features and a combination of the 618 best features. Similar performances were obtained for larger networks. For example, an ANN with four hidden layers with 20, 11, 15 and 7 neurons improved the RMSE by less than 0.1 % for wavelet transforms. It was decided to use the same architecture for each type of features to facilitate the comparison. Compared to object classification, good performances for smaller networks have been reported for other applications involving textural features [47, 14].

3 Results

3.1 Entropy

Different number of bins were used to calculate the local entropy. Fig. 11 shows the RMSE on the percentage passing for number of bins between 2 and 256. When the RMSE for all sieves is considered, the best performances were obtained for 8 bins (RMSE = 4.8 %). The optimal number of bins can vary for specific sieves. For example, for the finest sieve, the lowest RMSE (3.5 %) was

obtained with 4 bins. Except for 256 bins, better performances were obtained systematically for the finest sieve. This was also observed for the other types of features.

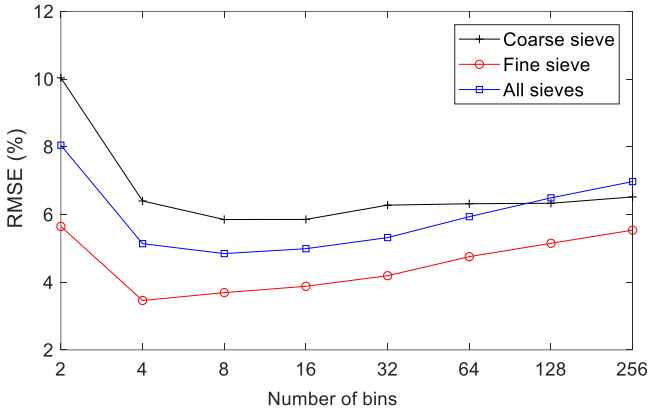

Fig. 11 RMSE on the percentage passing for the fine sieve, coarse sieve, and all sieves using local entropy with different bin sizes

3.2 LBP

To find the best LBP parameters, different radius values (1 to 12 pixels) and number of neighbors (4, 6, 8, 12, 16 pixels) were tested. By changing these parameters, the number of LBP features varied between 15 and 243. **Error! Reference source not found.** shows some examples of RSME for the percentage passing for all sieves combined. The two curves show the influence of the number of neighbors and the radius. The radius was set to 3 for the neighbor curve and the number of neighbors was set to 16 for the radius curve. It can be seen that better performances were obtained for higher numbers of neighbors. The radius curve shows that the minimum RMSE was obtained for an optimal radius of 3 pixels.

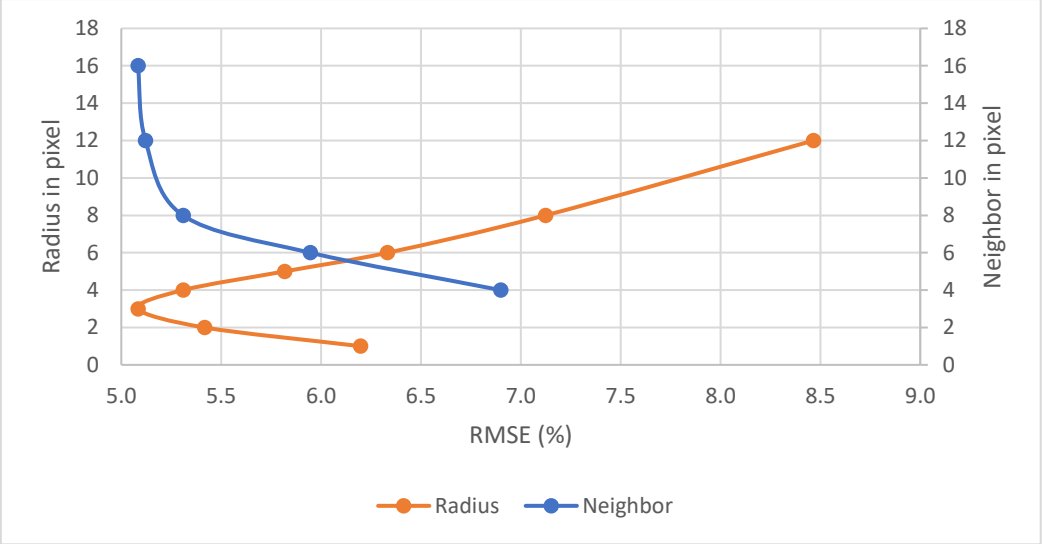

Fig. 12 RMSE on the percentage passing for LBP with different radius values and number of neighbor pixels for all sieve sizes.

Increasing the number of features through the number of neighbors or a combination of features for different LBP parameters had a relatively small influence on the RMSE. For example, the RMSE decreased from 6.5 to 4.1 % when the number of features was increased from 15 (radius of 2 pixels with 4 neighbors) to 3880 (all features combined). On the other hand, a selection of the best features coud lead to optimized results with a limited number of features. For instance, combining the features corresponding to 4 neighbors and radius values from 1 to 3 pixels, a total of only 45 features, led to a RMSE of 5.1 %. This accuracy is similar to the results for 12 neighbors and a radius of 3 pixels (133 features), and 16 neighbors and a radius of 3 pixels (243 features).

3.3 LCP and CLBP

As with LBP, the RMSE for LCP and CLBP decreased with an increasing number of neighbors. **Error! Reference source not found.** shows the RMSE for all sieve sizes as a function of the number of neighbors (*N*) and the radius (*R*) for LCP and CLBP. As with LBP, the best results for LCP when considering all sieve sizes were obtained for a radius of 3 pixels. The lowest RMSE (4.8 %) for LCP was obtained for 12 neighbors and a radius of 3 pixels. A RMSE of 3.8 % was obtained for a

combination of all features with 1 to 12 neighbors and radius values of 1 to 3 pixels. For CLBP, the best result were obtained for 5 neighbors and a radius of 2 pixels (RMSE of 5.3 %). A RMSE of 3.8 % was obtained when combining all CLBP features.

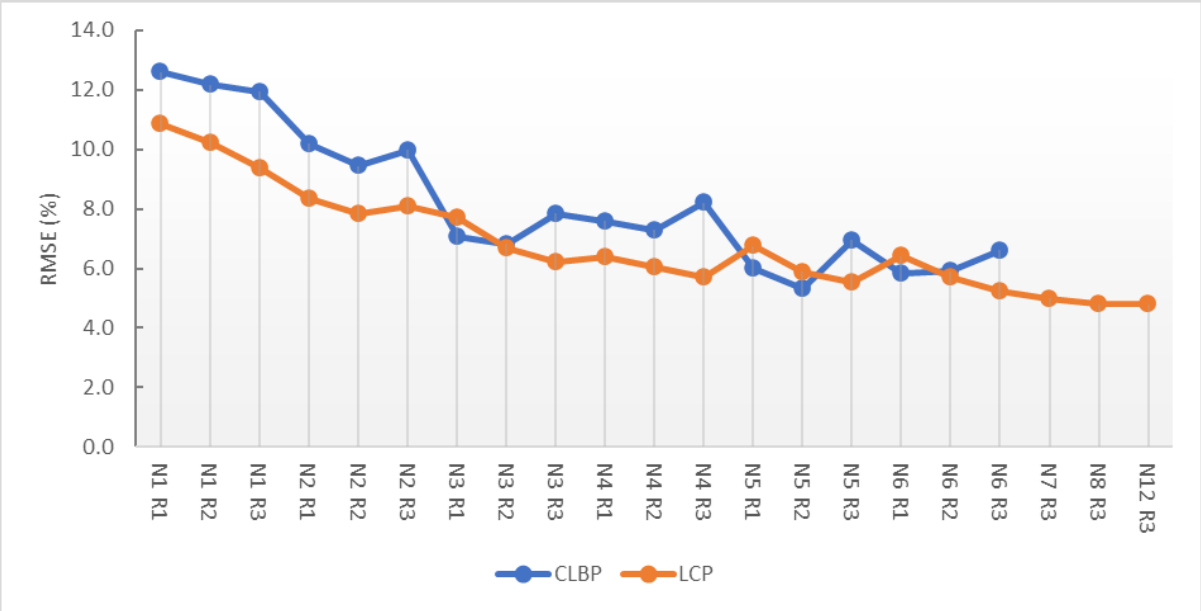

Fig. 13 RMSE of LCP and CLBP for different numbers of neighbors and radius for all sieve sizes. *N* refers to the number of neighbors and *R* to the radius

3.4 Fourier Transform

Error! Reference source not found. presents the results obtained using the mean and standard deviation of the Fourier transform magnitude spectrum for 5 rings with size multipliers between 0.5 and 2.4. The size multiplier corresponds to the ratio between the radiuses of each consecutive ring. When all sieves are considered, the lowest RMSE was 5.8 % for a size multiplier of 1.6. This ratio is roughly equal to the ratio between the opening size of consecutive sieves (e.g. 710 µm / 425 µm = 1.67). The best results for the finest sieve (RMSE = 1.9 %) were obtained for a size multiplier of 1.

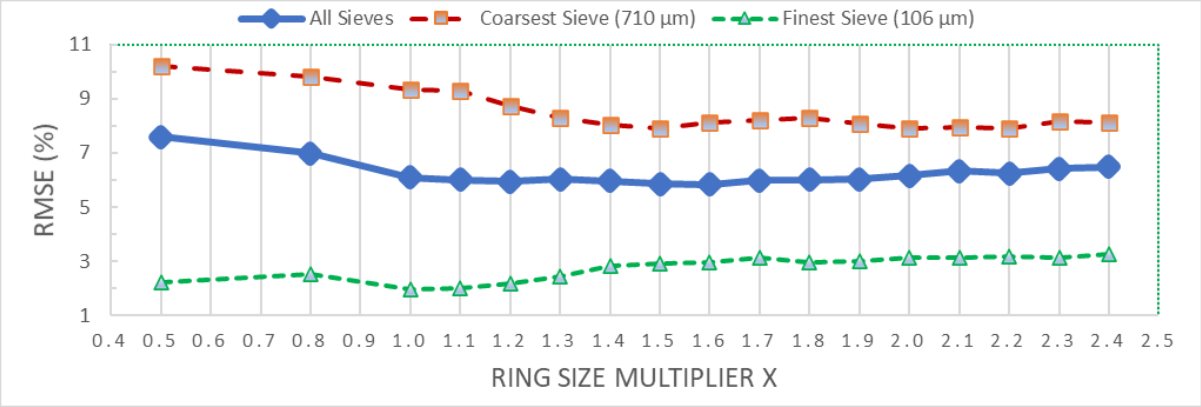

Fig. 14 RMSE results for the mean and standard deviation of the Fourier transform magnitude spectrum of 5 rings with size multipliers from 0.5 to 2.4

The influence of the number of rings was verified for a size multiplier of 1. The number of rings was found to have relatively little influence on the predicted PSD. For instance, when varying the number of rings from 5 to 20, the RMSE only varied between 6.9 and 7.1 %. Using a large number of rings, as Yaghoobi et al. [23] who used 20 equal rings to predict the size of fragmented rock, does not significantly increase the accuracy of the PSD determination.

3.5 Other methods and combination of all features

Error! Reference source not found. compares the RMSE of each type of features when predicting the percentage passing for the finest and coarsest sieves, and all sieves. For LBP, LCP, and CLBP, the methods with the best results, RMSE are presented for both the best set of parameters and a combination of all features. For each method, better results were obtained for fine particles. The combination of all LCP features (LCP ALL) provided the best results for fine particles with a RMSE of 1.9 %. The combination of all CLBP features (CLBP ALL) provided the best results for coarse particles with a RMSE of 5.2 %. LCP ALL and CLBP ALL both obtained the lowest RMSE for all sieves (3.8 %). The RMSE for the coarsest sieve, finest sieve, and all sieves were respectively 4.7, 1.7, and 3.4 % for a combination of the 618 best features from all methods. This set of features combined Fourier Equal Ring (68), Wavelet (49), HOG (12), Fourier Ring size Multiplier (20), Haralick (56), CLBP (7), Gabor (84), LBP (135), LCP (169), and Entropy (18).

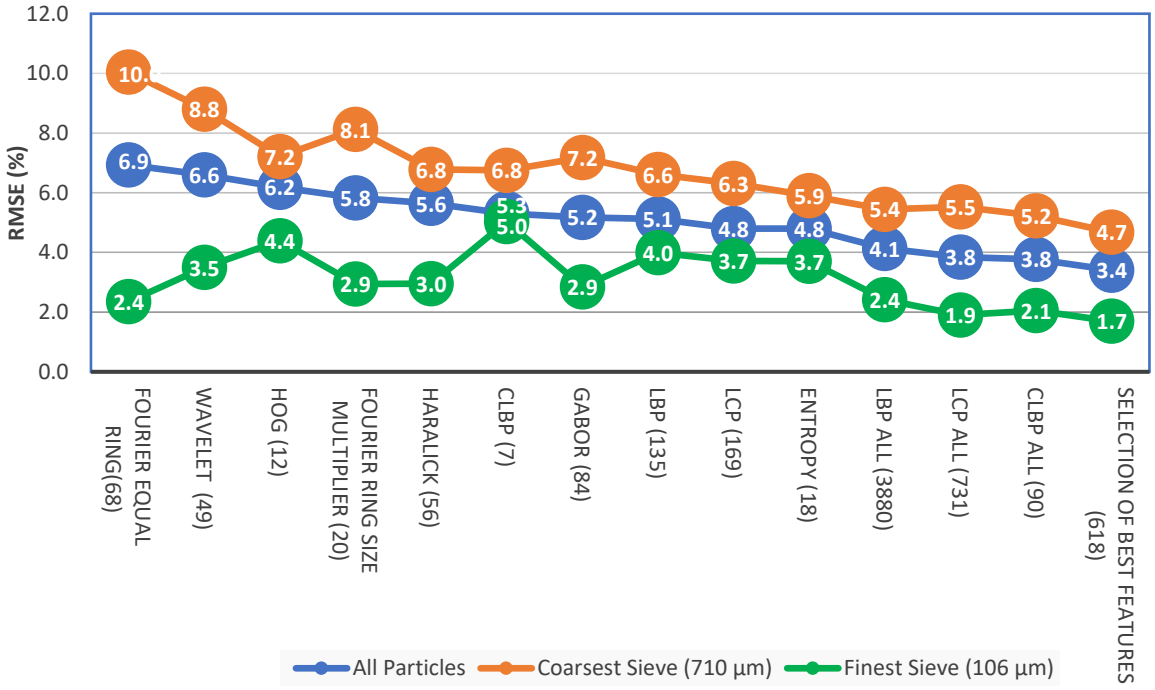

Fig. 15 Comparison of the RMSE on the percentage passing for each type of extracted features and some selected combinations. The number of features for each method is given between parentheses

Irrespective of the type of textural feature and sieve size, the RMSE of 1.7 to 10 % in Fig. 16 are low compared to the RMSE of the experimental results presented in the literature. For instance, the comparison of FragScan, WipFrag and Split by Liu and Tran [11] resulted in RMSE varying between 13 and 36 % for dry waste rock. RMSE of 15 and 20 % were achieved by WipFrag and Fragalyst when comparing automatic segmentation with manual particle contouring [9]. On the other hand, it should be realized that the synthetic soil photographs (Fig. 1) are idealized and probably much easier to handle than real soil photographs with varying illumination, particle colour, and particle shape.

Fig. **Error! Reference source not found.** compares the real and predicted percentages passing for each image and each sieve for the combination of the 618 best features. The overall coefficient of determination (R^2) is 0.99. As noted previously, larger sieve sizes are associated with poorer predictions. The RMSE gradually increases from 1.7 to 4.7 % with increasing sieve sizes. The R^2 coefficient also decreases from 0.99 to 0.91 between the 106 and 710 µm sieves.

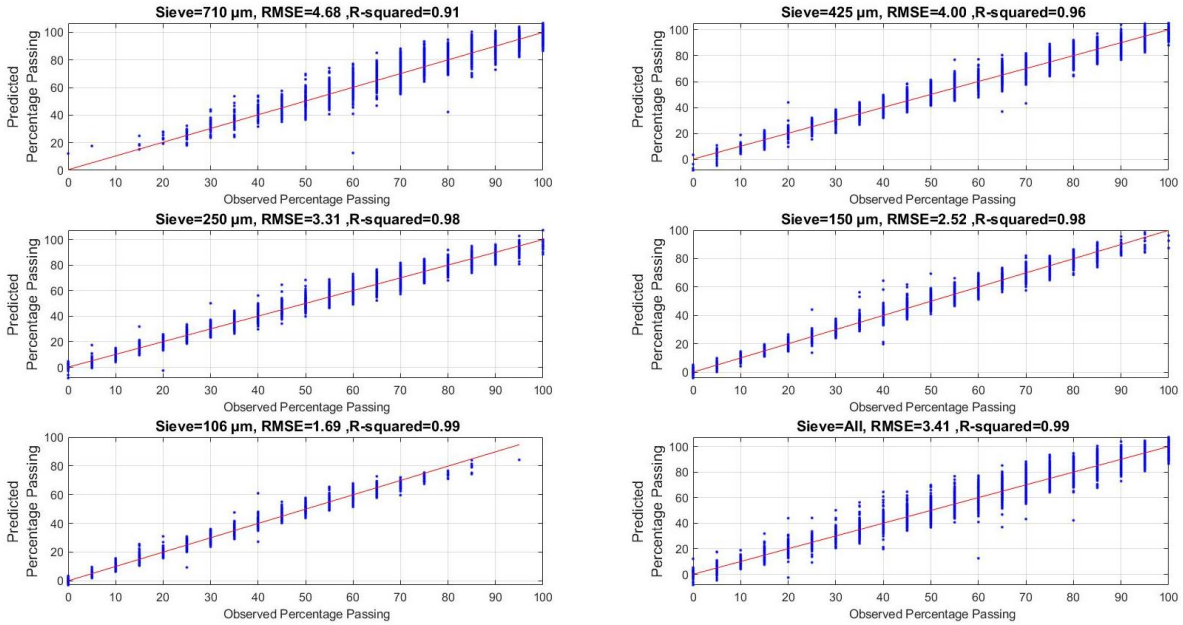

Fig. 16 Comparison of the real and predicted percentages passing for a selection of the 618 best features

Error! Reference source not found. compares some randomly selected examples of real and predicted PSD for the 618 best features. The percentage passing is plotted on the usual semi log plot. Blue diamonds and lines show the real PSD and red circles and lines indicate the predictions. The RMSE for these examples varied between 1.4 and 5.2 %. Overall, the real and predicted PSD are similar, even for a RMSE of 5.2%.

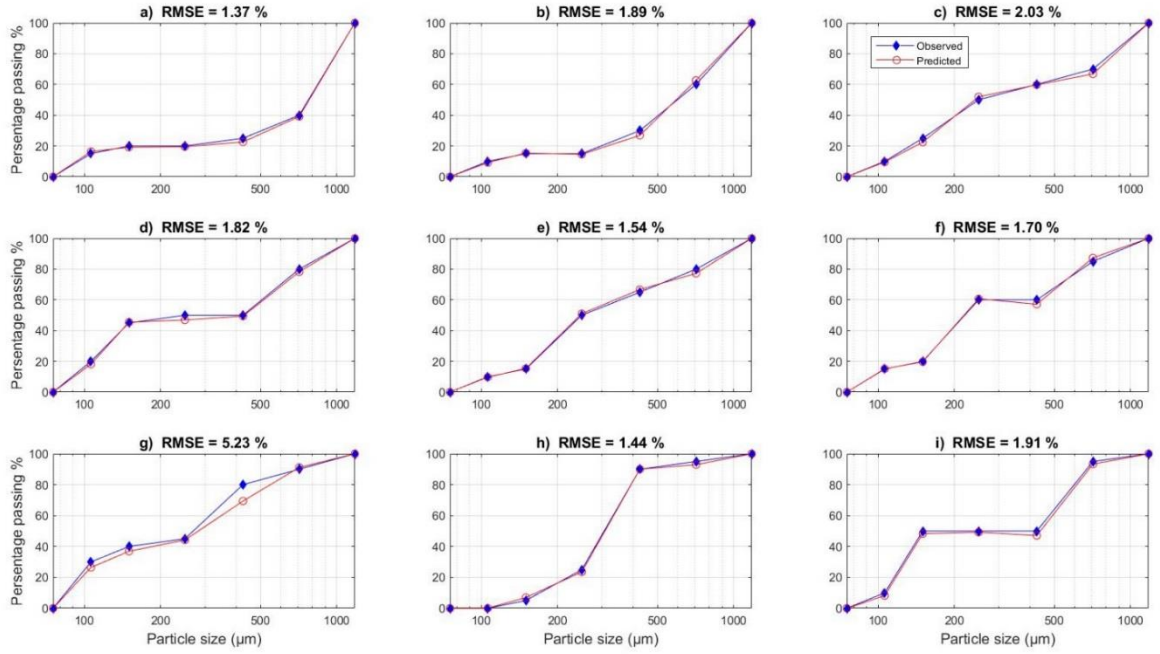

Fig. 17 Examples of real and predicted PSD for the synthetic images obtained with the DEM and the 618 best features

The error percentage on D_{50} was calculated for the combination of 618 features. The D_{50} for both the predicted and observed PSD were calculated based on the method presented by Thyabat et al. [48]. The D_{50} error for the combination of all methods is 6.1 %. The predicted and observed D_{50} values are compared in **Error! Reference source not found.**. As with the RMSE on the percentages passing, the error on the D_{50} is low compared to the values presented in the literature for real photographs. For example, Liu and Tran (1996) obtained D_{50} errors of approximately 55 % with Split and WipFrag. Again, the good performances for this dataset were expected considering its idealized nature.

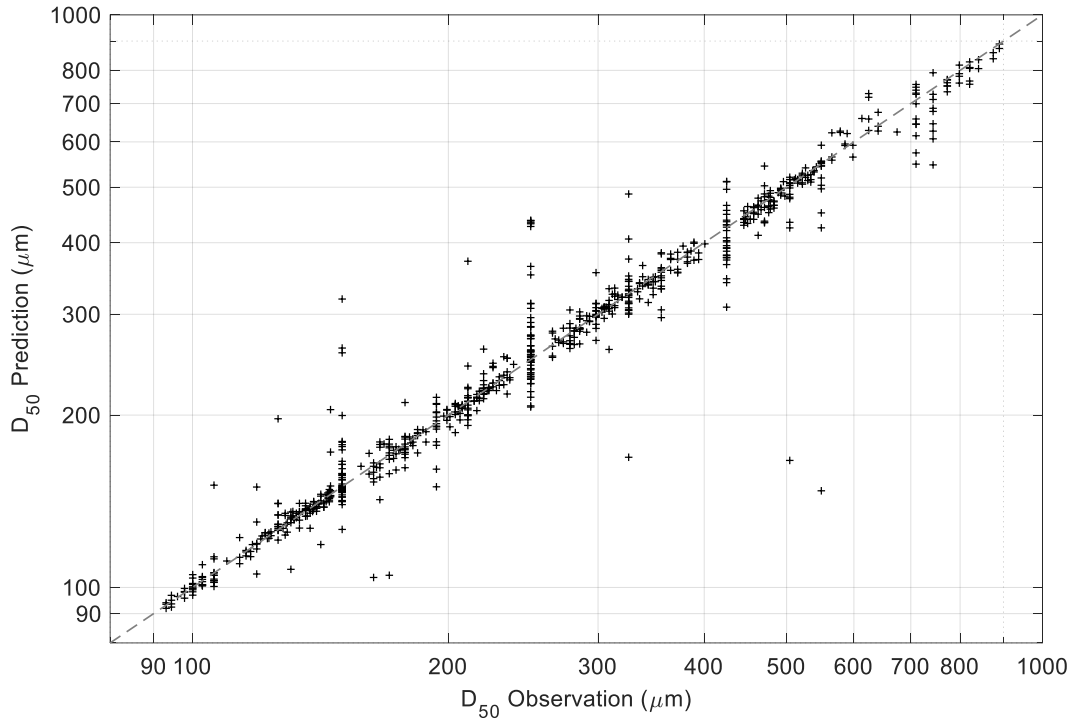

Fig. 18 Prediction and Observation of D_{50} for the 618 best features

In some cases, the predicted D_{50} were not accurate because of the shape of the PSD. An example of such PSD is shown in **Error! Reference source not found.**i. The flat part of the curve corresponding to a percentage passing of 50 % implies that the DEM specimen does not contain particles between 150 and 425 µm. Half of the specimen mass corresponds to particles that are larger than 425 µm than while the other half comprises particles that are smaller than 150 µm. This renders the D_{50} undefined. These cases correspond to the vertical lines in Fig. 18 at observed D_{50} values corresponding to sieves sizes of 150, 250 and 425 µm.

4 Conclusion

This paper introduced a dataset containing 53003 pairs of synthetic photographs of granular material. This dataset was used to train a series of ANN to predict the PSD. A series of textural features were used as the ANN inputs. All features were able to predict the PSD separately or in combinations with a good accuracy. Methods based on the concept of Local Binary Pattern (LBP), such as LCP and CLBP, could predict the PSD better than the other methods, and could be used independently. RMSE of 4.1%, 3.8%, and 3.8% were obtained for LBP, LCP and CLBP, respectively. To our best knowledge, this paper presented the first example of PSD determination with these methods. The best results were obtained with a combination of 618 features. The RMSE for the coarsest sieve, finest sieve, and all sieves were respectively 4.7 %, 1.7 %, and 3.4 %. The relative error on the D_{50} was 6.1 %.

These errors are small compared to previously published results due to the idealized nature of the images in the dataset. The next step is to test these methods with real datasets. Other applications can also be envisioned. For example, similar methods could be used for soil classification (e.g., ASTM D2487, ASTM International 2020), and the estimation of soil properties (e.g., shear strength, compressibility, compaction, hydraulic conductivity).

Synthetic image datasets like the one that was introduced in this paper offer several advantages. As they are easier to generate than real photograph datasets, they allow various parameters to be tested rapidly. For example, the influence of lighting on the performance of different methods could be tested by adding a randomly positioned light source before the image rendering. Synthetic images could also be used to increase the size of real photograph datasets.

Acknowledgements The authors gratefully acknowledge the support of Hydro-Québec and NSERC for this project.

References

1. ASTM C136 / C136M-19 (2019) Standard test method for sieve analysis of fine and coarse aggregates. ASTM International, West Conshohocken, PA. https://10.1520/C0136_C0136M-19

2. Bardet J -P (1997) Experimental soil mechanics. Upper Saddle River, NJ: Prentice Hall

3. ASTM D2487-17e1 (2017) Standard practice for classification of soils for engineering purposes (Unified Soil Classification System). ASTM International, West Conshohocken, PA. https://10.1520/D2487-17E01

4. Tuceryan M, Jain AK (1993) Texture analysis. Handbook of pattern recognition and computer vision. World Scientific, pp 235–276. https://doi.org/10.1142/9789814343138_0010

5. Maerz NH, Palangio TC, Franklin JA (1996) WipFrag image based granulometry system. Proc FRAGBLAST 5:91–99

6. Guyot O, Monredon T, LaRosa D, Broussaud A (2004) VisioRock, an integrated vision technology for advanced control of comminution circuits. Miner Eng 17:1227–1235. https://doi.org/10.1016/j.mineng.2004.05.017

7. Chavez R, Cheimanoff N, Schleifer J. (1996) Sampling problems during grain size distribution measurements. In: Proceedings of the fifth international symposium on rock fragmentation by blasting-FRAGBLAST. pp 245–252

8. Thurley MJ (2011) Automated online measurement of limestone particle size distributions using 3D range data. J Process Control 21:254–262. https://doi.org/10.1016/j.jprocont.2010.11.011

9. Sudhakar J, Adhikari GR, Gupta RN (2006) Comparison of fragmentation measurements by photographic and image analysis techniques. Rock Mech Rock Eng 39:159–168. https://doi.org/10.1007/s00603-005-0044-9

10. Raina AK, Choudhury PB, Ramulu M, Chakraborty AK, Misra DD (2002) Fragalyst-An indigenous digital image analysis system for grain size measurements in mines. J Geol Soc India 59:561–569. http://cimfr.csircentral.net/id/eprint/999

11. Liu Q, Tran H (1996) Comparing systems-validation of fragscan, wipfrag and split. Meas Blast Fragm Franklin Katsabanis 151–155

12. Cheimanoff NM, Chavez R, Schleifer J (1993) Fragscan: A scanning tool for fragmentation after blasting. In: International symposium on rock fragmentation by blasting. pp 325–330

13. Girdner KK, Kemeny JM, Srikant A, McGill R (1996) The split system for analyzing the size distribution of fragmented rock. Meas Blast Fragm Franklin Katsabanis 101–108

14. Liu L, Chen J, Fieguth P, Zhao G, Chellappa R, Pietikäinen M (2019) From BoW to CNN: Two decades of texture representation for texture classification. Int J Comput Vis 127:74–109. https://doi.org/10.1007/s11263-018-1125-z

15. Tian DP (2013) A review on image feature extraction and representation techniques. Int J Multimed Ubiquitous Eng 8:385–396. https://www.earticle.net/Article/A202352

16. Haralick RM, Shanmugam K, Dinstein IH (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 610–621. https://doi.org/10.1109/TSMC.1973.4309314

17. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24:971–987. https://doi.org/10.1109/TPAMI.2002.1017623

18. Manjunath BS, Ma W-Y (1996) Texture features for browsing and retrieval of image data. IEEE Trans Pattern Anal Mach Intell 18:837–842. https:// doi.org/10.1109/34.531803

19. Itoh H, Matsuo K, Oida A, Nakashima H, Miyasaka J, Izumi T (2008) Aggregate size measurement by machine vision. J Terramechanics 45:137–145. https://doi.org/10.1016/j.jterra.2008.09.001

20. Paclík P, Verzakov S, Duin RPW (2005) Improving the maximum-likelihood co-occurrence classifier: a study on classification of inhomogeneous rock images. In: Scandinavian Conference on Image Analysis. Springer, pp 998–1008. https://doi.org/10.1007/11499145_101

21. Hryciw RD, Ohm H-S, Zhou J (2015) Theoretical basis for optical granulometry by wavelet transformation. J Comput Civ Eng 29:4014050. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000345

22. Shin S, Hryciw RD (2004) Wavelet analysis of soil mass images for particle size determination. J Comput Civ Eng 18:19–27. https://doi.org/10.1061/(ASCE)0887-3801(2004)18:1(19)

23. Yaghoobi H, Mansouri H, Farsangi MAE, Nezamabadi-Pour H (2019) Determining the fragmented rock size distribution using textural feature extraction of images. Powder Technol 342:630–641. https://doi.org/10.1016/j.powtec.2018.10.006

24. Raza Ansari S (2018) Application of Machine Learning Techniques for Soil Type Classification of Karanataka. PhD dissertation, National College of Ireland. Dublin, 2018. http://norma.ncirl.ie/id/eprint/3443

25. Ghalib AM, Hryciw RD, Shin SC (1998) Image texture analysis and neural networks for characterization of uniform soils. In: Computing in Civil Engineering. ASCE, pp 671–682

26. Shirazi HHA, Rahimi Rahbar R, Saghatoleslami N (2004) Modeling of texture and color froth characteristics for evaluation of flotation performance in Sarcheshmeh copper pilot plant, using image analysis and neural networks. Int J Eng 17:121–130

27. Ko Y-D, Shang H (2011) A neural network-based soft sensor for particle size distribution using image analysis. Powder Technol 212:359–366. https://doi.org/10.1016/j.powtec.2011.06.013

28. Hamzeloo E, Massinaei M, Mehrshad N (2014) Estimation of particle size distribution on an industrial conveyor belt using image analysis and neural networks. Powder Technol 261:185–190. https://doi.org/10.1016/j.powtec.2014.04.038

29. Pirnia P, Duhaime F, Manashti J (2018) Machine learning algorithms for applications in geotechnical engineering. In Geo Edmonton: the 71st Canadian Geotechnical Conference and the 13th Joint CGS/IAH-CNC Groundwater Conference.

30. Pirnia P, Duhaime F, Ethier Y, Dubé J-S (2019) ICY: An interface between COMSOL multiphysics and discrete element code YADE for the modelling of porous media. Comput Geosci 123:38–46. https://doi.org/10.1016/j.cageo.2018.11.002

31. Gadkari D (2004) Image quality analysis using GLCM. Master's thesis, University of Central Florida. http://purl.fcla.edu/fcla/etd/CFE0000273

32. Monzel R (2020) haralickTextureFeatures. MATLAB Central File Exchange.

https://www.mathworks.com/matlabcentral/fileexchange/58769-haralicktexturefeatures

Retrieved December 24, 2020.

33. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE, pp 886–893. https://doi.org/10.1109/CVPR.2005.177

34. Pang Y, Yuan Y, Li X, Pan J (2011) Efficient HOG human detection. Signal Processing 91:773–781. https://doi.org/10.1016/j.sigpro.2010.08.010

35. Math Works (2019) Extract histogram of oriented gradients (HOG) features. https://www.mathworks.com/help/vision/ref/extracthogfeatures.html. Accessed 26 December 2020.

36. González RC, Woods RE (2018) Digital image processing; 4th Edition. Pearson, New York.

37. Larkin KG (2016) Reflections on shannon information: In search of a natural information-entropy for images. arXiv Prepr arXiv160901117

38. Sudha D, Ramakrishna M (2017) Comparative study of features fusion techniques. In: 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT). IEEE, pp 235–239. https://doi.org/10.1109/ICRAECT.2017.39

39. Kaya Y, Ertuğrul ÖF, Tekin R (2015) Two novel local binary pattern descriptors for texture analysis. Appl Soft Comput 34:728–735. https://doi.org/10.1016/j.asoc.2015.06.009

40. Deng L, Yu R (2015) Pest recognition system based on bio-inspired filtering and lcp features. In: 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, pp 202–204. https://doi.org/10.1109/ICCWAMTIP.2015.7493975

41. Guo Y, Zhao G, Pietikäinen M (2011) Texture classification using a linear configuration model based descriptor. In: BMVC. Citeseer, pp 1–10. http://dx.doi.org/10.5244/C.25.119

42. Guo Z, Zhang L, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. IEEE Trans image Process 19:1657–1663. https://doi.org/10.1109/TIP.2010.2044957

43. Guo Y, Zhao G, PietikäInen M (2012) Discriminative features for texture description. Pattern Recognit 45:3834–3843. https://doi.org/10.1016/j.patcog.2012.04.003

44. MATLAB Code. Center for Machine Vision and Signal Analysis, University of Oulu. https://www.oulu.fi/cmvs/node/33019.

45. Szeliski R (2010) Computer vision: algorithms and applications. In: Texts in Computer Science, Springer Science & Business Media. https://doi.org/10.1007/978-1-84882-935-0

46. Ohm HS, Hryciw RD (2014) Soil Fabric Characterization by Wavelet Transformation of Images. In: Geo-Congress 2014: Geo-characterization and Modeling for Sustainability. pp 723–730. https://doi.org/10.1061/9780784413272.070

47. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386

48. Al-Thyabat S, Miles NJ, Koh TS (2007) Estimation of the size distribution of particles moving on a conveyor belt. Miner Eng 20:72–83. https://doi.org/10.1016/j.mineng.2006.05.011