Grain Size Analyses of Granular Material Based on Image Analysis Techniques and Machine Learning

by

MohammadJavad MANASHTI

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph. D.

MONTREAL, MAY 24, 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. François Duhaime, Thesis Supervisor
Département de génie de la construction, École de technologie supérieure

Mr. Christian Desrosiers, President of the Board of Examiners
Département de génie logiciel et des TI, École de technologie supérieure

Mr. Jean-Sébastien Dubé, Member of the jury
Département de génie de la construction, École de technologie supérieure

Mr. Fernando Salazar González, External Evaluator
International Centre for Numerical Methods in Engineering

THIS THESIS WAS PRENSENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

MAY 24, 2022

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Analyses granulométriques par techniques d’analyse d’images et apprentissage machine

MohammadJavad MANASHTI

RÉSUMÉ

La détermination de la distribution granulométrique (PSD) est un test clé en génie civil et géotechnique. Le tamisage est la méthode standard. On utilise cette méthode des milliers de fois par jour dans le monde entier. Le tamisage prend du temps. Il est bruyant et produit de la poussière. Il consomme également de l'eau et peut potentiellement être influencé par les erreurs des opérateurs. Le dimensionnement des particules par techniques d’analyse d’images est de plus en plus courant. La plupart des méthodes commerciales sont basées sur la segmentation d'images. La PSD est déterminée sur la base de la distribution de taille des zones segmentées. Cette thèse est centrée sur le développement de nouvelles méthodes basées sur les textures et les réseaux de neurones. Elle présente une étude comparative des performances des méthodes traditionnelles d'extraction de caractéristiques et des réseaux de neurones convolutifs (ConvNet). Deux jeux de données ont été utilisés : un jeu de données synthétique contenant 53103 paires d'images synthétiques produites avec le code d'éléments discrets YADE, et un jeu de données réel contenant 9600 photographies de 15 matériaux.

Le projet comportait trois volets principaux. Dans la première partie, neuf techniques traditionnelles d'extraction de caractéristiques ont été évaluées avec le jeu de données synthétiques. Les caractéristiques comprenaient celles de Haralick, l’histogramme de gradients orientés, l’entropie locale, les motifs binaires locaux et les paramètres dérivés, la transformée rapide de Fourier, les filtres de Gabor et la transformée en ondelettes discrète de Haar. Dans la deuxième partie, le modèle ConvNet PSDNet a été développé, affiné et testé sur le même jeu de données synthétiques. Des ConvNets pré-entraînés ont également été évalués pour l'extraction de caractéristiques et l'apprentissage par transfert. Les ConvNet pré-entraînés qui ont été testés dans cette recherche comprennent AlexNet (Krizhevsky, Sutskever, and Hinton, 2012), SqueezeNet (Iandola et al., 2016) , GoogLeNet (Szegedy et al., 2015), InceptionV3 (Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna, 2016), DenseNet201 (Huang, Liu, Van Der Maaten, and Weinberger, 2017), MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, and Chen, 2018), ResNet18 (Wu, Zhong, and Liu, 2018), ResNet50 (He, Zhang, Ren, and Sun, 2016) , ResNet101 (He et al., 2016), Xception (Chollet, 2016), InceptionResNetV2 (Ioffe and Szegedy, 2015; Szegedy, Ioffe, Vanhoucke, and Alemi, 2016), ShuffleNet (Zhang, Zhou, Lin, and Sun, 2017), and NasNetMobile (Zoph, Vasudevan, Shlens, and Le, 2018). Dans la troisième partie, les extracteurs de caractéristiques traditionnels, PSDNet et les ConvNet pré-entraînés ont été évalués avec les photographies réelles de deux manières. D'abord, le jeu de données a été divisé en sous-ensembles de données réguliers d’entrainement de validation et de test. Deuxièmement, un matériau a été retiré du jeu de données pendant l’entrainement et la

validation pour vérifier la capacité des ConvNet à prédire la PSD de matériaux que le réseau n'a jamais vu. Le matériau qui a été retiré a été utilisé comme jeu de données test.

Nos résultats montrent que chacune des techniques comparées peut conduire à de bonnes prédictions de la PSD, autant pour les jeux de données réels que synthétiques. Pour le jeu de données synthétiques, la valeur de la racine carrée de l'erreur quadratique moyenne (RMSE) sur les pourcentages passants est de 3,4 % lors de l'utilisation d'une sélection des 618 meilleures caractéristiques traditionnelles comme entrée pour un réseau de neurones artificiels. La meilleure version de PSDNet a produit une valeur de RMSE sur les pourcentages passants de 2,8%. InceptionResNetV2 et DenseNet201 ont donné les meilleurs résultats pour l'extraction de caractéristiques et l'apprentissage par transfert, respectivement, avec des RMSE de 3,6 %. Pour les photographies, PSDNet a atteint des RMSE sur les pourcentages passants de 3,7 et 5,5 % pour les images en couleur et en tons de gris, respectivement. Les meilleures performances pour l'apprentissage par transfert et l'extraction de caractéristiques des modèles pré-entraînés ont été obtenues pour NasNetLarge et ResNet101 avec un RMSE de 4,3 et 4,0%, respectivement. Les meilleurs résultats ont été obtenus en utilisant InceptionResNetV2 comme méthode d'extraction de caractéristiques basée sur l'apprentissage par transfert, avec une valeur RMSE de 1,7 %. De plus, des résultats similaires ont été obtenus en utilisant toutes les caractéristiques extraites de gris et de couleur PSDNet composées de 1000 composants, avec un RMSE de 1,8 %.

Des résultats significativement meilleurs ont été obtenus avec PSDNet lors de l'utilisation d'images en couleur au lieu d'images en tons de gris. De meilleurs résultats ont également été obtenus pour les particules fines, en particulier pour les techniques traditionnelles d'extraction de caractéristiques. Une combinaison d'images capturées du haut et du bas donne de meilleurs résultats pour la prédiction de la PSD en comparaison avec l'utilisation de la vue de dessus seule. Pour les photographies, de bons résultats ont été obtenus dans le test de retrait de matériau avec une valeur minimum de RMSE de 1,5%. Des valeurs de RMSE beaucoup plus élevées (> 31,8 %) ont été obtenues pour les deux PSD extrêmes (matériaux le plus grossier et le plus fin). Ce résultat met en évidence le fait que les méthodes qui ont été comparées dans cette thèse ne peuvent pas être utilisées pour extrapoler en dehors de la plage PSD utilisée dans le jeu de données d'apprentissage.

Cette étude est la première à introduire l'utilisation de ConvNet pré-entraînés pour la prédiction de la PSD par l'extraction de caractéristiques et l'apprentissage par transfert. L'entraînement avec un jeu de données réel est une différence clé entre la détermination de la PSD avec des réseaux de neurones et les méthodes classiques de traitement d'images. L'apprentissage permet de prendre en compte implicitement les particules cachées. Les logiciels commerciaux basés sur la segmentation des particules doivent tenir compte explicitement de ces particules cachées à l'aide de méthodes statistiques. La capacité d'apprentissage des ConvNet permet également d'améliorer le modèle pendant son fonctionnement grâce à l'extension du jeu de données. Le jeu de données synthétiques utilisé dans cet article est considéré comme un moyen prometteur de créer de grands jeux de données comprenant à la fois des photographies réelles et des images de synthèse.

Mots-clés: Réseaux de neurones convolutifs, Distribution granulométrique, Éléments discrets, Analyse d'image, Image synthétique, Extraction de caractéristiques, PSDNet, ConvNet pré-entraîné, Apprentissage par transfert, Réseau de neurones artificiels

Grain Size Analyses of Soils Based on Image Analysis Techniques and Machine Learning

MohammadJavad MANASHTI

ABSTRACT

The determination of the particle size distribution (PSD) is a key test in geotechnical and civil engineering. Sieving is the standard method. It is performed all around the world thousands of times every day. Sieving is time-consuming. It is noisy and it produces dust. It also consumes water, and it retains the potential for human operators' mistakes. Recently, image-based techniques for particle sizing have become more common. Most commercial methods are based on image segmentation. The PSD is determined based on the size distribution of the segmented areas. This thesis is centered on the development of new methods based on textures and neural networks. It presents a comparative study of the performances of traditional feature extraction methods and convolutional neural networks (ConvNet). Two datasets were used: a synthetic dataset containing 53103 pairs of synthetic images produced with the discrete element code YADE, and a real dataset containing 9600 photographs of 15 materials.

The thesis has three main parts. In the first part, nine traditional feature extraction techniques were evaluated with the synthetic dataset. The features included Haralick features, histogram of oriented gradients, local entropy, local binary pattern, local configuration pattern, complete local binary pattern, fast Fourier transform, Gabor filter, and Haar discrete wavelet transform. In the second part, PSDNet, a ConvNet model, was developed, fine-tuned, and tested on the same synthetic dataset. Pretrained ConvNets were also evaluated for feature extraction and transfer learning. The pretrained ConvNet that were tested in this research include AlexNet (Krizhevsky et al., 2012), SqueezeNet (Iandola et al., 2016) , GoogLeNet (Szegedy et al., 2015), InceptionV3 (Szegedy, Vanhoucke, et al., 2016), DenseNet201 (Huang et al., 2017), MobileNetV2 (Sandler et al., 2018), ResNet18 (Wu et al., 2018), ResNet50 (He et al., 2016) , ResNet101 (He et al., 2016), Xception (Chollet, 2016), InceptionResNetV2 (Ioffe and Szegedy, 2015; Szegedy, Ioffe, et al., 2016), ShuffleNet (Zhang et al., 2017), and NasNetMobile (Zoph et al., 2018). In the third step, traditional feature extractors, PSDNet, and pretrained ConvNet were evaluated on the real photograph dataset in two ways. First, the dataset was split into the regular training/validation/test subdatasets. Second, one material was removed completely during training and validation to check the ability of ConvNet to predict the PSD of materials that the network has never seen. The unseen material was used as the test dataset.

Our findings indicate that each of the compared techniques can lead to good predictions of the PSD for both real and synthetic datasets. For the synthetic dataset, the Root Mean Square Error (RMSE) value on the percentages passing was 3.4 % when using a selection of the 618 best traditional features as the input for an artificial neural network. The best version of PSDNet resulted in a RMSE value on the percentages passing of 2.8%. InceptionResNetV2 and DenseNet201 gave the best results for feature extraction and transfer learning, respectively, with the same RMSE values of 3.6 %. For the real photographs, PSDNet achieved a RMSE on the percentages passing of 3.7 and 5.5 % for the color and grayscale band, respectively. The best performances for pretrained models' transfer learning and feature extraction were obtained for NasNetLarge and ResNet101 with a RMSE of 4.3 and 4.0 %, respectively. The best results were obtained by using InceptionResNetV2 as the transfer-learning-based feature extraction method, with an RMSE value of 1.7 %. As well, similar results were obtained by using all PSDNet gray and color extracted features consisting of 1000 components, with an RMSE of 1.8%.

Significantly better results were achieved when using color images in PSDNet instead of grayscale images. Better results were also obtained for the fine particles in the PSD, especially for traditional feature extraction techniques. A combination of views of the soil captured from the top and bottom yields better results for PSD prediction than using only the top view. For the real photographs, good results were achieved in the material removal test with a minimum RMSE of 1.5 %. Much higher RMSE values (> 31.8 %) were obtained for the two extreme PSD (coarsest and finest). This highlights the fact that the methods that were compared in this thesis cannot be used to extrapolate outside of the PSD range used in the training dataset.

This study was the first to introduce pre-trained ConvNet for feature extraction or transfer learning methods of PSD prediction. The training with a real dataset is a key difference between PSD determination with neural networks and classical image processing methods. Training allows hidden particles to be taken into account implicitly. Commercial software based on particle segmentation must account for these hidden particles explicitly using statistical methods. The capacity of ConvNet to learn also allows the model to be enhanced during its operation through the extension of the dataset. The synthetic dataset that was used in this paper is seen as a promising way to build large datasets including both real photograph and synthetic images.

Keywords: Convolutional neural network, Particle size distribution, Discrete elements, Image analysis, Synthetic image, Feature extraction, PSDNet, Pretrained ConvNet, Transfer learning, Artificial neural network

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

ANN	Artificial Neural Network
PSD	Particle Size Distribution
ConvNet	Convolutional Neural Network
RMSE	Root Mean Square Error
R^2	Coefficient of Determination
PSDNet	Particle Size Distribution Network
ReLU	Rectified Linear Units
GLCM	Gray Level Cooccurrence Matrix
DEM	Discrete Element Method
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
LCP	Local Configuration Pattern
CLBP	Completed Local Binary Pattern
CNTK	Microsoft Cognitive Toolkit
pyDGS	Python Implementation of the Wavelet Feature Extraction Method
UAV	Unmanned Aerial Vehicles
T	Top (Image View)
U	Under (Image View)
TU	Top and Under (Image View)
STU	Stretched Top and Under (Image View)
CU	Color Images from Under
GSTU	Grayscale Images with the Stretched Top and Under Views
GPU	Graphics Processing Unit
SGDM	Stochastic Gradient Descent with Momentum
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
NSERC	National Sciences and Engineering Research Council of Canada
MiC	Microscopic Configuration
Fitnet	Function Fitting Neural Networks

LIST OF SYMBOLS

C_u	uniformity coefficient
D_{60}	grain sizes corresponding to percentages passing of 60%
D_{50}	grain sizes corresponding to percentages passing of 50%
D_{30}	grain sizes corresponding to percentages passing of 30%
D_{10}	grain sizes corresponding to percentages passing of 10%
C_c	coefficient of curvature
Rc	center section ratio
Re	mean ratio of four sections around the center
μ	mean
σ	standard deviation
N_g	number of gray levels
P_j	probability of occurrence of gray level
$H(X)$	entropy of the neighborhood X around a center pixel
x_p	gray levels for the neighboring pixels
$\mathrm{s}(x_p - x_c)$	step function
$F(u, v)$	two-dimensional discrete Fourier transform
$F(x, y)$	two-dimensional Gabor function
S	Gaussian function standard deviations
Ψ	mother wavelets
A	approximation coefficients
H	horizontal coefficients
V	vertical coefficients
D	diagonal coefficients
$\mathrm{f}(x)$	threshold function

CHAPTER 1

INTRODUCTION

1.1 Introduction

The most common geotechnical tests for granular soils, such as sand and gravel, concern the determination of their particle size distribution (PSD). Sedimentation and sieving (Gee and Or, 1986), the current standard methods, were introduced in the first half of the 20th century. These methods are precise but time-consuming. Commercial laboratories would benefit from more rapid methods to determine the PSD.

In geotechnical engineering, PSD are commonly used for soil classification (Raza Ansari, 2018), material selection for road and embankment dam construction, and soil property estimation (e.g. permeability). The majority of civil engineering projects require PSD tests. Therefore, thousands of tests are completed every day. PSD are utilized in several other fields, including space exploration (Carrier, 2003), aerosol (Herman, Browning, and Reagan, 1971), mineral industries (Patel, Chatterjee, and Gorai, 2019), flotation (Saghatoleslam, Karimi, Rahimi, and Shirazi, 2004), and nanoparticles (Caputo, Clogston, Calzolai, Rösslein, and Prina-Mello, 2019).

Image analysis techniques and machine learning can provide faster methods for PSD determination. For testing and evaluating these methods, a labelled soil-image dataset is needed. For this reason, two large image datasets were used in this thesis: a dataset of synthetic soil particles containing 53003 top and bottom views of spherical particles deposited in a virtual box (Figure 1.1), and a dataset of real soil photographs captured in the laboratory containing 9600 top views (Figure 1.2).

Figure 1.1 shows three samples of synthetic soil particle images prepared with YADE. The top row (a, b, and c) represents the top views and the bottom row (d, e, and f) represents the bottom views of fine (a and d), widely graded (b and e), and coarse (c and f) materials.

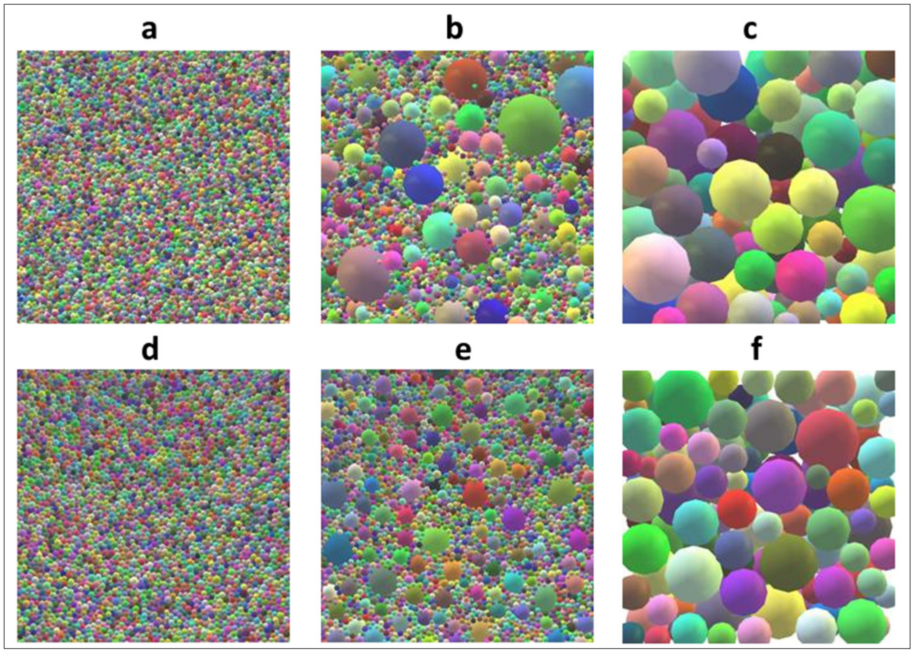

Figure 1.1 Synthetic granular material images prepared with YADE. The top row (a, b, and c) represents the top views, and the bottom row (d, e, and f) represents the bottom views of fine (a and d), widely graded (b and e), and coarse (c and f) particles

Figure 1.2 shows three real granular material photographs with clear PSD differences. Figure 1.2a shows a material with a relatively uniform PSD. Figure 1.2b and 1.2c show two more widely graded materials with coarser particles. The actual percentages passing for this dataset were determined by sieving.

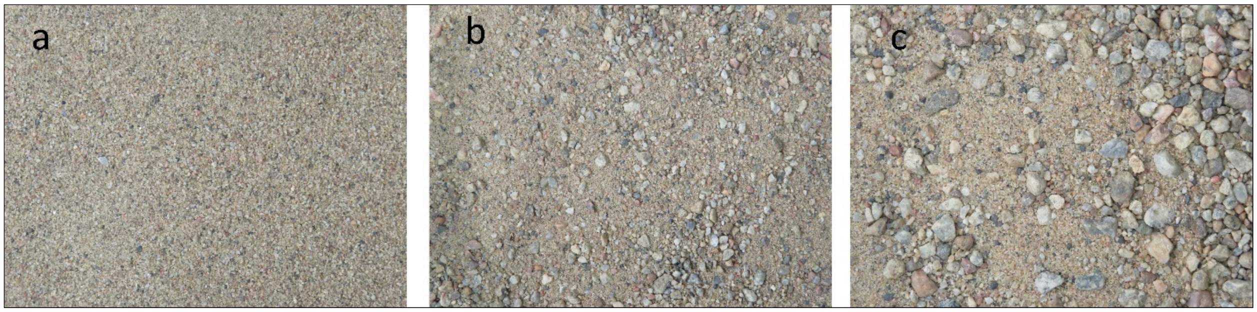

Figure 1.2 Real soil images. (a) soil with a relatively uniform PSD, (b) and (c) more widely graded soil particles (coarser particles)

New methods based on image analysis need to be validated and compared with the current standard methods. Two methods, traditional feature extraction and convolutional neural networks (ConvNet), are evaluated in this thesis to predict soil PSD based on images. Nine traditional feature extraction methods were compared, including Haralick features, histogram of oriented gradients, local entropy, local binary pattern, local configuration pattern, complete local binary pattern, fast Fourier transform, Gabor filter, and Haar discrete wavelet transform. ConvNet, a new type of neural network for inputs consisting of images, was used for the first time in geotechnical engineering to predict the PSD of granular materials. Moreover, ConvNet models were evaluated for feature extraction and transfer learning methods.

Image-based methods for PSD determination can be faster and less expensive than the current standard methods, such as sieving. The potential for human error can be reduced. The measurement accuracy can be increased by sampling a larger volume of material. Because of their low energy and water consumption, these techniques are more environmentally friendly. They can be repeated several times rapidly and without touching the samples. They can also be used with drones.

1.2 Objectives

The main objective of this thesis was to develop a series of methods to determine the PSD with image analysis techniques and ConvNets. The thesis follows three secondary objectives. First, the study aimed to evaluate traditional feature extractors to determine the size of synthetic granular materials. To this end, a dataset prepared with the discrete element code YADE was used to compare nine different feature extraction methods. Neural networks were used to predict the PSD from the extracted features. For the second objective, PSDNet, a new ConvNet, was developed to determine the PSD of synthetic images of granular materials. The performances of pre-trained ConvNet models for PSD determination with feature extraction or transfer learning methods were evaluated. For the third objective, the performances of PSDNet, pre-trained ConvNets, and traditional feature extractors were compared for PSD prediction based on real photographs of granular materials. A dataset consisting of 9600 photographs of 15 different materials was used. In addition to PSDNet, nine traditional feature extraction methods and 15 pre-trained ConvNets models were evaluated and compared. Moreover, all the methods were used to predict the PSD for a soil that was not shown to the networks during the training phase.

1.3 Outline and content

Chapter 2 presents a literature review with the following subjects:

- Particle size classification and distribution curve;
- Sieving and its disadvantages;
- New methods to determine soil PSD;
- Review of PSD determination systems;
- Image analysis techniques;
- Particle image datasets.

The thesis contains three articles in Chapters 3, 4, and 5. These three chapters can be summarized as follows:

Chapter 3 addresses the performances of traditional feature extraction techniques to predict the PSD of granular materials. Nine methods were evaluated, including Haralick features, histogram of oriented gradients, local entropy, local binary pattern, local configuration pattern, complete local binary pattern, fast Fourier transform, Gabor filter, and Haar discrete wavelet transform. Furthermore, this article introduces a dataset of synthetic images of an idealized granular material with spherical particles. This dataset was created with the discrete element code YADE. Different tuning methods are detailed to find the optimum parameters in MATLAB for the feature extraction techniques.

Chapter 4 addresses PSD predictions for granular materials with ConvNets. Different ConvNet models were tested to create PSDNet, the optimum model. PSDNet was evaluated using a series of datasets based on the synthetic images introduced in Chapter 3. Images of different sizes and color bands were compared. Performances for predictions based on pre-trained

ConvNet models, including AlexNet, SqueezeNet, GoogLeNet, InceptionV3, DenseNet201, MobileNetV2, ResNet18, ResNet50, ResNet101, Xception, InceptionResNetV2, ShuffleNet, and NASNetMobile, were evaluated.

Chapter 5 evaluates the performances of PSDNet to determine the PSD from real soil photographs. The dataset includes 9600 images of 15 granular materials. PSDNet was evaluated with different image sizes (32, 64, 96, 128, 160, and 192 pixels), and with both grayscale and color images. Nine traditional feature extraction methods and 15 pre-trained ConvNets models were evaluated. Pre-trained ConvNets were used with feature extraction and transfer learning methods. The ability of PSDNet to predict the PSD of materials that have never been seen by the networks was also evaluated.

Chapter 6 presents a discussion of the results and recommendations for future works.

CHAPTER 2

LITERATURE REVIEW

This chapter first describes the concepts of particle size classification and particle size distribution (PSD). Sieving and its disadvantages are then presented. The following sections reviews other methods used for PSD determination, including image analysis techniques with emphasis on convolutional neural networks (ConvNets). The last section addresses the methods used for generating synthetic and real photograph datasets.

2.1 Particle size classification

Soil and granular material in general contain particles of various sizes and shapes. Particle size allows soil particles to be classified into several size groups. Standard soil classification systems, including BS 1377 (1990), AASHTO (2009), and USCS (ASTM D2487, 2017), classify particles based on their size into boulders, cobbles, gravel, sand, silt, and clay categories (Bardet, 1997). Figure 2.1 shows the boundaries between the categories for some important engineering soil classification systems. ASTM D422 (2007) classifies particles in the following way: silt size, 0.074 to 0.005 mm, clay size, smaller than 0.005 mm, and colloids, smaller than 0.001 mm. This standard is no longer in use. It has been replaced by two new standards for PSD determination: ASTM D6913 (2019b) and ASTM D7928 (2016). The size limits for silt, clay, and colloids are not defined in these new standards.

Most of the particles encountered in civil engineering projects are covered in Figure 2.1. Still, some projects involve blocks that are larger than 1000 mm. For example, Bouchard (2016) characterized the rockfill material at the Romaine-2 dam. The PSD for this material includes particles that are larger than 1200 mm. According to Hydro-Québec's internal practices, such PSD analysis requires excavation in the compacted material. The size distribution of particles larger than 80 mm is determined in the field, while the distribution of particles smaller than 80 mm is determined in the laboratory. The entire sample's sieving is done at the site on an 80 mm square mesh sieve. Then, for the fraction less than 80 mm, a sample of about 45 kg is taken for particle size analysis in the laboratory. The mass determination for particles between 80 and 300 mm is carried out using a construction scale in the field. For particles larger than 300 mm, the mass is determined using a mechanical shovel or loader with a system to assess the load handled by the machine. Figure 2.2 shows the measurement and weighting of large blocks (Bouchard, 2016). The advantages of using image-based methods to determine the PSD for these coarse materials are obvious. Using photographs to determine the PSD avoids having to manipulate several tons of material.

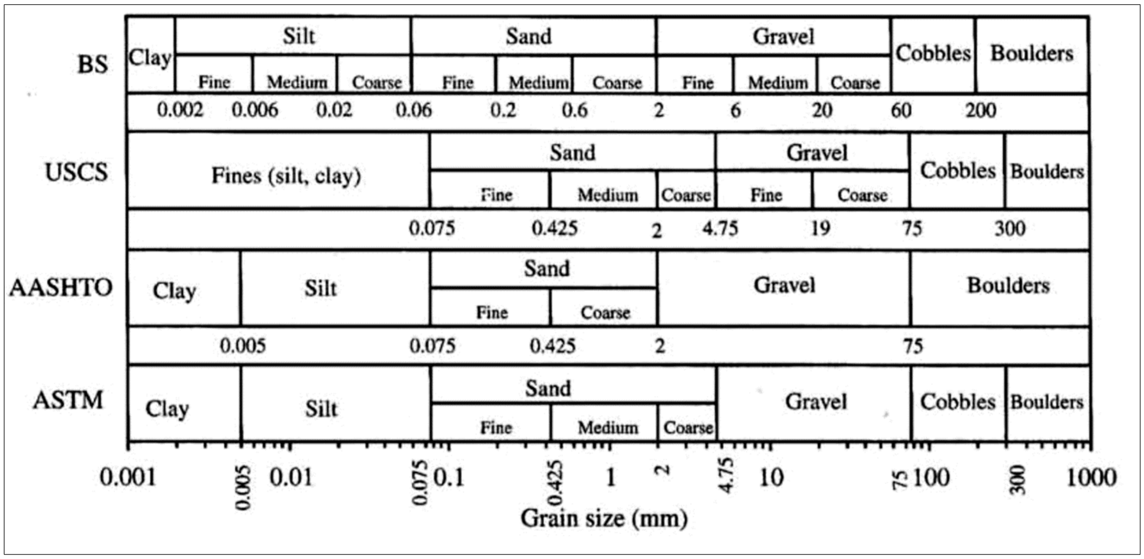

Figure 2.1 Particle size classification in the BS 1377, USCS, AASHTO, and ASTM D422 soil classification systems. Taken from Bardet (1997)

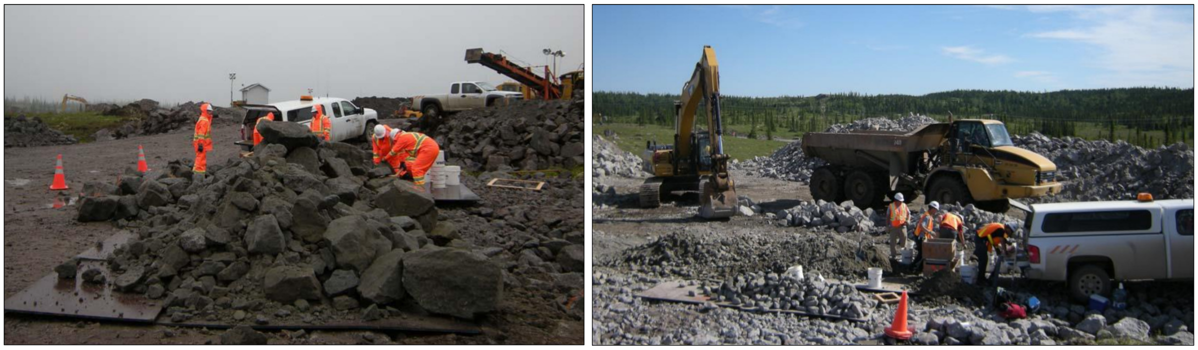

Figure 2.2 Carrying out a particle size analysis in the field. The left image shows how to measure the blocks, and the right image shows how to weigh them. Taken from Bouchard (2016)

2.2 Particle size distribution curve

The PSD of a soil illustrates the mass percentage of soil finer than a given size presented on a logarithmic scale (Figure 2.3). The percentage passing must necessarily increase with particle size because it shows a cumulative distribution. The PSD gives much information on the engineering properties of soils. The position of the curve specifies the fineness or coarseness of the soil. The slope and general shape of the curve give information on the grading of the PSD and the presence of gaps in the particle size range.

Several parameters are used to describe the PSD. The two most common parameters are the uniformity and curvature coefficients. The uniformity coefficient (C_u) gives information on the slope of the PSD:

$$C_u = \frac{D_{60}}{D_{10}} \tag{2.1}$$

where D_{10} and D_{60} and are the grain sizes corresponding to percentages passing of 10 and 60 %, respectively (Figure 2.3). C_u illustrates the mean slope of the PSD between percentages passing of 10 and 60 %. The shape of the PSD can also be described by the coefficient of curvature (C_c):

$$C_c = \frac{D_{30}^2}{D_{10}\ \ D_{60}} \tag{2.2}$$

where D_{30} is the size corresponding to a percentage passing of 30 % (Figure 2.3). Soils with $C_u > 4$ (gravel) or $C_u > 6$ (sand), and $1 < C_c < 3$ are described as “well graded.” Other soils are described as “poorly graded.” Poorly-graded soils comprise uniform and gap-graded soils.

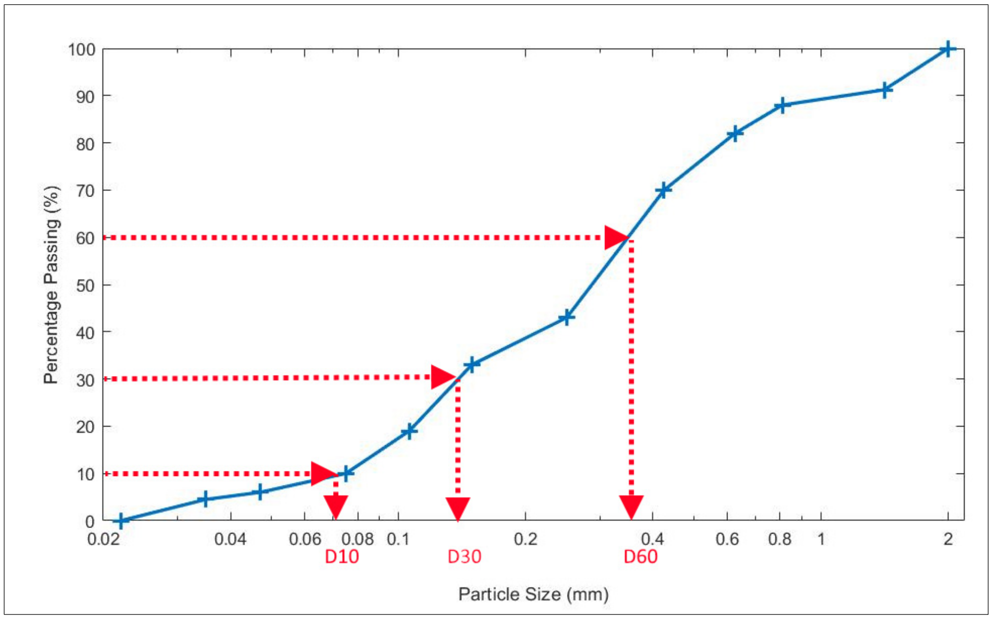

Figure 2.3 Particle size distribution curve

2.3 Sieving

Aside from the determination of the water content, sieving is considered the most common soil test in geotechnical engineering. Sieving and Atterberg limit tests are two essential elements for classifying soils. Sieving is also crucial in the concrete and pavement construction industries for quality control auditing of aggregate materials (Ohm, 2013).

The most comprehensive standard for mechanical sieving is ASTM D6913 (2019b) . It applies to particles larger than 0.075 mm. Sieving uses a stack of sieves placed in order of decreasing screen opening size from top to bottom. The soil sample is fed into the top sieve, and shaken in the vertical and horizontal directions to accelerate the soil grain passage through the stack. The grain morphology can also affect the sieving results, as their shortest length determines the soil grain sizes (Jung, 2009).

Ohm (2013) divided the disadvantages of sieving into four categories. First, test duration is the main problem. Sieving requires a considerable amount of time, up to several days depending on the range of particle size and number of manipulations involved (e.g. washing on the 75-

µm sieve, separation on the 2-mm or 5-mm sieves, drying, etc.). Second, damage to the sieves and system maintenance can be costly. Sieves can be damaged during mesh cleaning and must be replaced regularly. The third category includes environmental concerns. Sieve shakers are energy-demanding systems. A large volume of water is also required for PSD determination. According to Ohm (2013), the water consumption per sample ranges from 7.56 to over 37.8 liters depending on the type of soil and the manipulation involved. The fourth category is centered on work environment issues. Sieving is associated with noise and vibration. These concerns are especially important when the sieve shakers are not isolated or placed in a separate room.

Even if sieving is considered the reference method for PSD determination, it is still associated with measurement errors. Clogging is a common problem. Although clogging increases for poorly-graded materials, it is still a source of error in materials with a wide PSD (Jung, 2009). Sampling errors can also be a problem (Dubé, Ternisien, Boudreault, Duhaime, and Éthier, 2021)

2.4 Other methods for PSD determination

Abbireddy and Clayton (2009) have reviewed the different methods for PSD determination. Available methods include hydrometer tests (0.2-75 µm), sieving (75-75,000 µm), electrical sensing zone (0.4-1,200 µm), laser diffraction (0.1-2,000 µm), single-particle optical sizing (1-5,000 µm), and image analysis (unlimited detection range depending on image quality and scale).

The hydrometer test (ASTM D7928) is used to determine the size distribution of fine-grained material with grains smaller than 75 µm and larger than 0.2 µm. Sedimentation results obtained with the hydrometer test are often combined with sieving results to determine the PSD of widely graded materials. Sedimentation is based on Stokes' Law (Stokes, 1891). Stokes' Law describes the fact that larger particles fall faster than smaller particles in a liquid. A hydrometer measures the density of a soil suspension based on the concept of buoyancy. A larger density implies a larger proportion of particles that are still in suspension. The grain size can be calculated based on the time and distance of particle sedimentation in the viscous fluid. Sedimentation analyses have several limitations based on the calculation. Falling particles are assumed to be spheres with similar specific gravity. The spheres are assumed to be separated, and not to interact during their fall (Bardet, 1997).

In the electrical sensing zone (ESZ) method (Vinet and Zhedanov, 2011), resistance variation is measured while particles pass through a small aperture. The particle volume in the aperture is proportional to the measured electrical impedance magnitude (Jackson, Logan, Alldredge, and Dam, 1995).

In the laser diffraction approach, particles are assumed to diffract light at an angle inversely related to their size (Zobeck, 2004). The Mie theory is used to determine the particle size. It relates the particle radius to the angular light intensity distribution (Wen, Aydin, and Duzgoren-Aydin, 2002). Individual particles are detected via a laser beam in the single-particle optical sizing approach while falling in the detection zone (White, 2003). The relationship between the intensity of light received, blocked, or scattered by detector and particle diameter is identified.

Techniques based on image analysis capture particle images by cameras with different methods (Prats-Montalbán, de Juan, and Ferrer, 2011). The photographs can be taken while the particles fall from a conveyor belt or are spread on a flat table. Furthermore, for inaccessible areas or coarse particles, images can be captured using UAVs and satellites. Image analysis methods will be described in more details in the next section.

2.5 Image analysis techniques

Thanks to higher efficiency in terms of energy and water requirements, quicker tests, and a cleaner process, image-based techniques for PSD determination have become more attractive in recent years (Ohm, 2013). Since image-based tests are shorter and involve less intricate procedures, they allow for a greater number of tests to be conducted in a given period, making testing more cost-effective. Accordingly, more samples mean a more comprehensive analysis and results with a higher statistical validity (Jung, 2009). Image analysis methods are also increasingly popular in domains that require fast methods to calculate or predict real-time PSD for decision-making, such as mining (Hamzeloo, Massinaei, and Mehrshad, 2014). Despite the advantages of image-based analysis systems, they still have a long way to go before they become the standard method for PSD analysis.

Image analysis methods can be divided into two groups: direct and indirect techniques. Direct methods are based on image segmentation. With this approach, the image is first segmented by identifying the contour of each particle. The size of the segmented areas is then determined. The statistical distribution of these sizes is then used to determine the PSD. WipFrag is an example of commercial software based on a direct method (Maerz, Palangio, and Franklin, 1996). Indirect methods extract textural features from the image and correlate them with particle size. With indirect methods, feature extraction techniques can be divided into traditional and modern methods. Traditional methods include statistical, local image pattern, and frequency domain methods. Modern feature extraction involve convolutional neural networks (Krizhevsky et al., 2012). Nine different traditional feature extraction techniques are discussed in Chapters 3 and 4. Convolutional neural networks, a novel feature extraction technique, is detailed in Chapters 4 and 5.

This chapter first describes segmentation methods and traditional feature extraction techniques, such as transform-based methods (Fourier transform, wavelet transform, and Gabor filter) and statistical methods (Haralick features). The next sections are centered on the role of artificial neural network and convolutional neural networks in image analysis and PSD determination.

2.5.1 Segmentation

Hamzeloo et al. (2014) described the segmentation of photographs of granular material on a conveyor belt. Some preprocessing steps were applied to the photographs before segmentation. As shown in Figure 2.4, these steps consist of image sharpening, edge extraction, and image thresholding. Examples of preprocessing steps and image segmentation are shown in Figure 2.5. First, the particle edges are detected (Figure 2.5b) and the image is converted to a binary image by thresholding (Figure 2.5c). Edges are associated with grey level contrasts. They can be identified through different gradient operators such as Sobel filters or more refined algorithms such as Canny edge detection (Gonzalez and Woods, 2018; Maerz et al., 1996; Sereshki, Hoseini, and Ataei, 2016). Next, closing filters like morphological dilation and erosion are applied to the images (Figure 2.5e) to obtain the region occupied by the particles. The area occupied by fine particles (Figure 2.5f) is identified by subtracting the segmented particles (Figure 2.5d) from the area occupied by the material (Figure 2.5e). The area occupied by small particles is used to obtain the percentage of fine particles in the PSD.

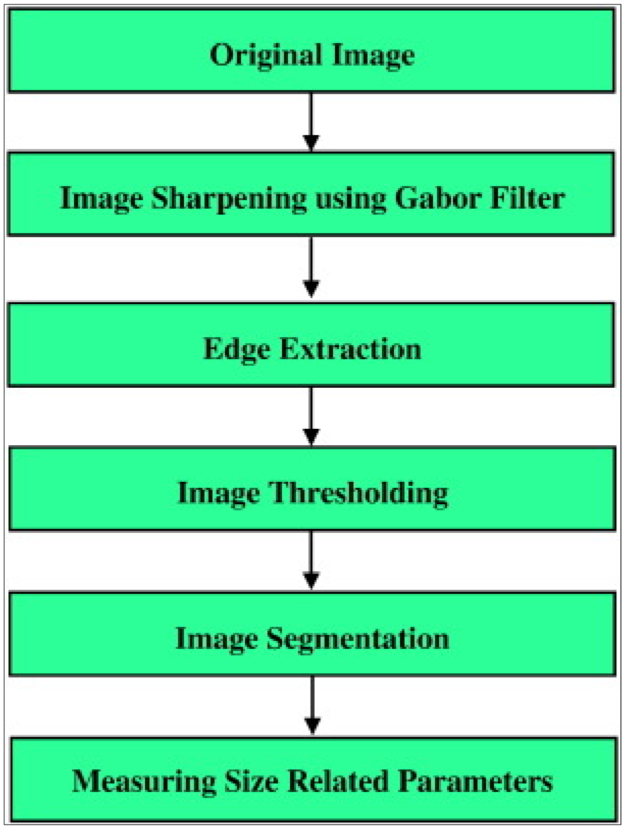
Original Image
Image Sharpening using Gabor Filter
Edge Extraction
Image Thresholding
Image Segmentation
Measuring Size Related Parameters

Figure 2.4 Procedure for images for segmentation and PSD determination. Taken from Hamzeloo et al. (2014)

PSD errors with segmentation methods vary widely depending on the parameters used to report the errors (e.g. D_{50}, percentage passing for specific particle size), the tested material (e.g. uniform, widely graded), and the reference method (e.g. sieving, manual image segmentation). Chapters 3 and 4 will review the error figures for the results that have been presented in the literature.

The main errors for PSD determination based on photographs can be described as sampling errors. In effect, photographs sample a surface of the granular material volume. Liu and Tran (1996) mentioned that photo analysis has three types of errors. First, it only processes the surface of the material. Second, small particles can be merged into coarse particles, and large particles can be divided into smaller ones. The last issue leads to an underestimation of fine particles. Tools based on image segmentation methods, such as WipFrag and Split Desktop (Potts and Ouchterlony, 2005; Split Engineering, 2021), cannot account for the particles that are hidden behind the particles that are seen by the camera (Ko and Shang, 2011a).

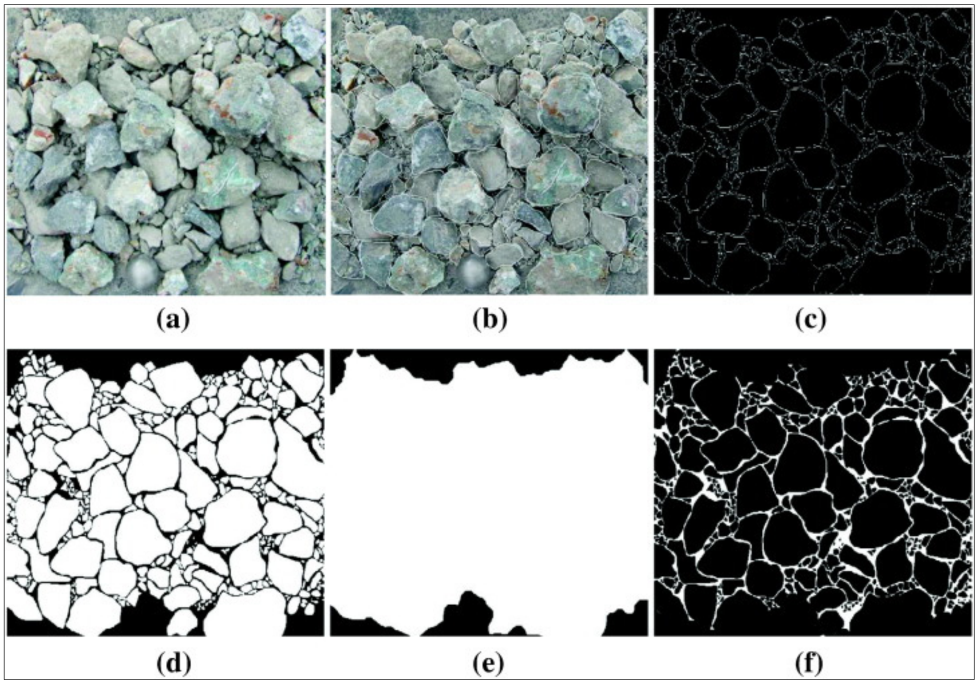

Figure 2.5 Segmentation of the original image (a) including edge extraction (b), conversion to a binary image (c), segmentation (d), determination of the region occupied by the sample (e) and fine material detection (f). Taken from Hamzeloo et al. (2014)

Yang and Chen (2017) used an image segmentation method to determine the PSD of particles lying on a conveyor belt or falling from a conveyor belt (Figure 2.6). They obtained a 3 % error on percentage passing for particles falling from the conveyor belt. When the particles were lying flat on the conveyor belt, the error increased to 12 %. This difference can be explained by the larger amount of hidden particles for samples that are lying on the conveyor belt. A larger proportion of particles can be captured by the camera when the sample is falling off the conveyor belt.

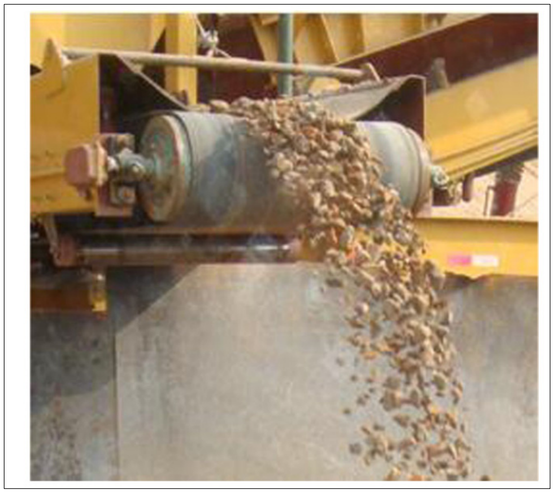

Figure 2.6 Particles falling from a conveyor belt. Taken from Yang and Chen (2017)

2.5.2 Traditional feature extractions

Methods based on traditional features can be divided into three main categories including statistical methods, local patterns, and transform-based methods (Tian, 2013). While statistical methods calculate statistics based on the spatial distribution of pixel intensities, local pattern methods describe the relationship between the gray level intensity in the neighborhood of each pixel. Transform-based methods extract features in the frequency domain. Features extracted with these methods can be correlated with the PSD. Detailed explanations are given in sections 3.2 and 3.3.2. Some examples of transform-based and statistical methods are presented in this section.

PSD determination methods are more commonly based on transform-based features. For example, Yaghoobi et al. (2019) used Fourier transforms, wavelet transforms, and Gabor filters, to determine the particle size of fragmented rock. Features were extracted from muck-pile images of 800×800 pixels. The mean and standard deviation of the Fourier magnitude spectrum was calculated for 20 ring filters representing different wavelengths. For wavelet analysis, the images were decomposed in five levels using Haar wavelets. 32 features were obtained with the mean and standard deviation of vertical, horizontal, and diagonal details for all decomposition levels, and the mean and standard deviation of the gray levels of the original image. For two-dimensional Gabor filters, a filter bank containing five scales and six directions was applied to the images. Using the Gabor filter for each image, the mean and standard deviation of 30 filtered images and the original image were calculated (62 features). The feature vectors were used individually or jointly as the input of a series of neural networks. The Split Desktop software (Potts and Ouchterlony, 2005; Split Engineering, 2021) was used to specify the fragmented rock size (ground truth). The best results were achieved for Fourier transforms, followed by Gabor filters, and a combination of Fourier and Wavelet transforms.

Shin and Hryciw (2004) evaluated two-dimensional Haar wavelet and Haralick features to discover the effect of illumination intensity and soil color on PSD determination. Two groups of laboratory experiments were conducted. In the first experiments, the impact of lighting intensity on textural features was studied. Four lighting intensities were used to capture 256×256 pixel photographs of the same material (Figure 2.7). A decreasing illumination intensity caused large differences for three specific Haralick features. The angular second moment was increased, whereas the contrast and variance were reduced. This influence was due to the more homogeneous gray levels associated with a lower illumination intensity. With the Haar wavelet transformation, each photograph was decomposed into eight levels. The energy at each decomposition stage increased while the images got lighter. This is due to the higher grayscale values associated with brighter images. In comparison with Haralick features, the wavelet method provided more reliable results because the energy normalization reduced the impact of the lighting intensity. In the second experiment, the influence of the soil particle color on the textural features and particle size determination were examined. Brighter sand with uniform light color and sand with darker particles and more contrast were selected for the comparison. The results show that the wavelet technique is relatively insensitive to variations in soil particle color.

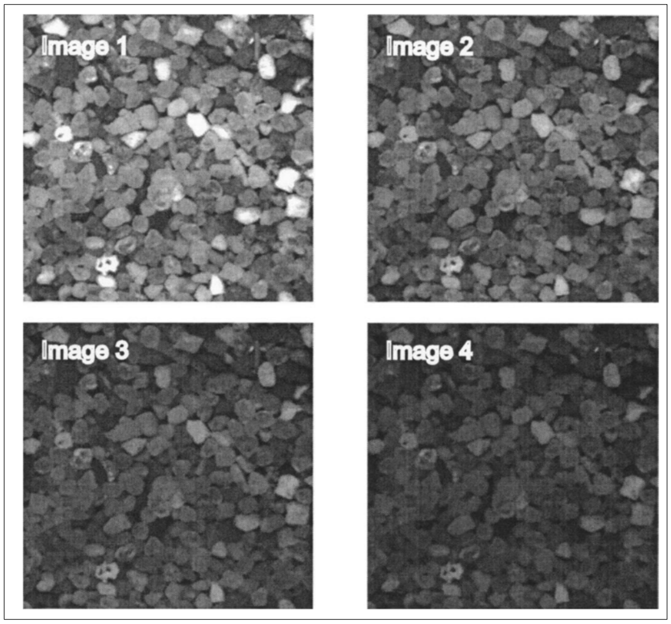

Figure 2.7 Images captured with four different lighting intensities on the same sample. Taken from Shin and Hryciw (2004)

Sedimaging is a PSD determination method that uses sedimentation in a water column to sort the particles according to their size to facilitate the image analysis (Ohm and Hryciw, 2014a). This method allows particle sizes in the 0.075-2.0 mm range to be determined. Image analysis is based on Wavelet transforms (Hryciw, Ohm, and Zhou, 2015). A DSLR camera was used to capture the image with a resolution of 0.020 mm. MATLAB was used to decompose each image into seven decomposition levels. The energy of each level was calculated. A relationship between the average number of pixels per particle diameter and the normalized energy for the seven decomposition levels (Wavelet index) was used to predict the particle size for regions of 128 × 128 pixels in the sedimentation column. The complete PSD was reconstituted by sorting the particle size for each region and by assuming that each region corresponds to the same weight percentage. An example of Sedimaging output is presented in Figure 2.8. Sedimaging produces more than 5,000 data points per PSD determination and, consequently, leads to continuous PSD curves. This is not the case with sieving which produces only one point per sieve. Compared with sieving, a root mean square error (RMSE) of 7.4 % can be calculated for the weight percentages from the results given in the paper for tests with nine different sands.

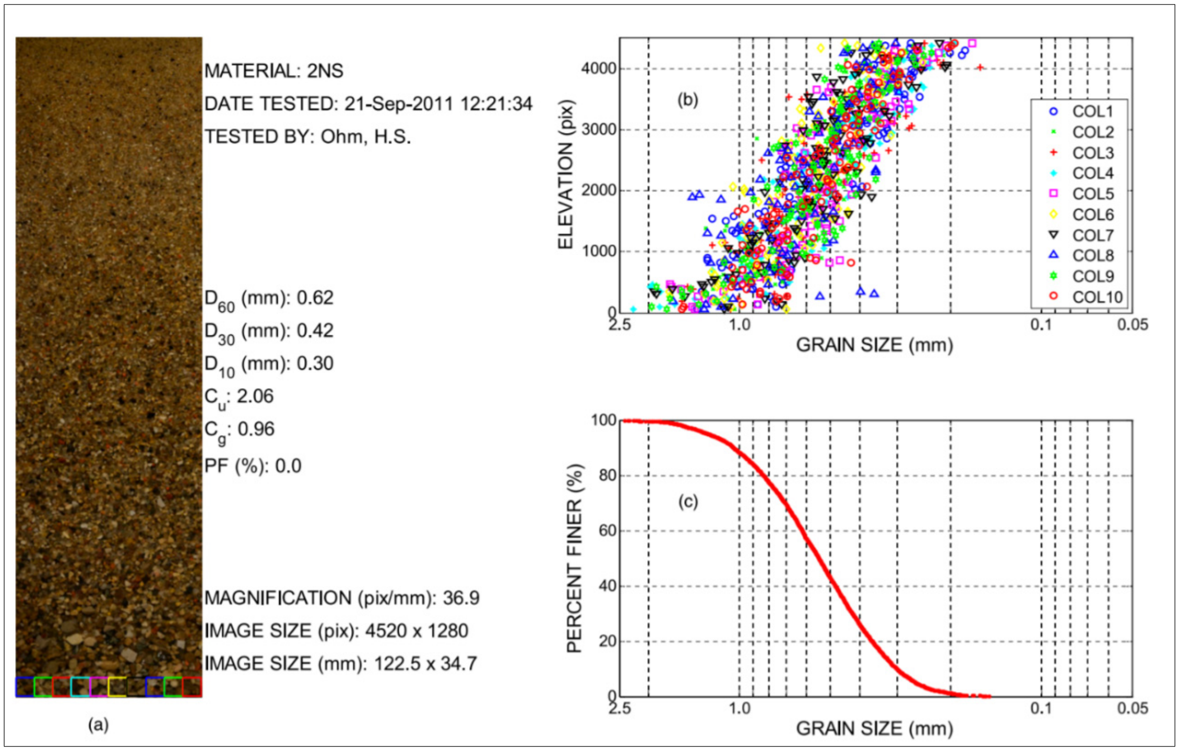

Figure 2.8 Typical Sedimaging results for one soil sample. (a) particles sorted based on their size, (b) decreasing particle size with rising elevation, and (c) PSD. Taken from Ohm and Hryciw (2014)

2.5.3 Image processing and neural networks

Artificial Neural Networks (ANN) form the basis of machine learning and deep learning. A simple example of ANN with five input nodes and two output nodes is shown in Figure 2.9. This simple network has one hidden layer with three neurons. Each node (or neuron) is connected to the nodes of the next layer. Each connection is associated with a weight and a threshold (or bias). The hidden layer is associated with a nonlinear transfer (or activation) function. The weight and threshold are determined during network training.

The structure of the network and training parameters are two important aspects of ANN. The number of hidden layers, number of neurons in each layer, and type of activation function are often chosen based on a combination of experience, and trial and error (Maren et al., 2014). Datasets must contain hundreds of samples for the training process.

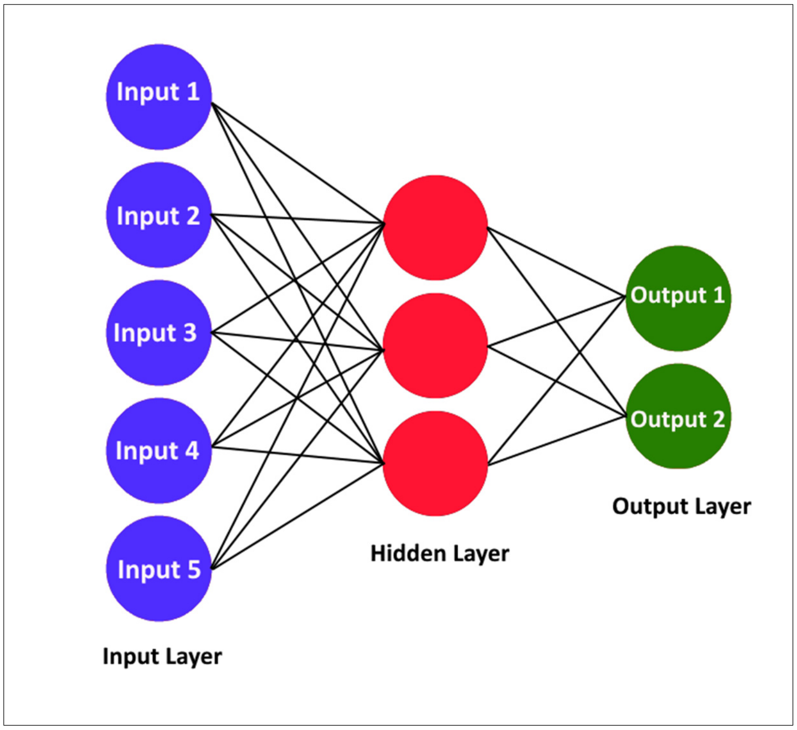

Figure 2.9 ANN example with five inputs, one hidden layer with 3 neurons, and two outputs

Some researchers have combined image processing methods with ANN for PSD determination (Ko and Shang, 2011a, 2011b, Hamzeloo et al. 2014, Yaghoobi et al., 2019). It is generally not possible to use image pixels directly as ANN inputs because of the number of weight parameters that would needed, the size of the database that would be needed to achieve proper generalization and to avoid overfitting, and the problem of vanishing gradients. Consequently, image preprocessing is often used to extract parameters from the images. Textural parameters that are related to the PSD can be used as ANN inputs. For example, Ko and Shang (2011a) defined a uniformity coefficient (*U*) used as an ANN input to predict particle sizes corresponding to mass percentage passing of 25, 50, and 75 %. Each image was divided into 5 sections. For each section, the ratio R of pixel values that were below a threshold was identified. Equation 1 defines *U* as:

$$U = \frac{|\mathrm{Rc} - \mathrm{Re}|}{Rc} \times 100 \tag{2.3}$$

Where *Rc* is the center section ratio, and *Re* is the mean ratio of four sections around the center (top, button, left, and right). When used as ANN input, *U* was able to predict particle size with high accuracy. WipFrag was used to determine the ground truth for PSD prediction. In Table 2.1, the comparison of three methods for predicting PSD is shown. As can be seen in the table, the neural network-based method has significantly better results.

Table 2.1 Summary of three methods R-squared PSD. Taken from Ko and Shang (2011a)

Models	R-Squared for responses (%)		
	D_{50}	D_{75}	D_{90}
Linear Regression	12.10	12.40	0.40
Quadratic	30.60	30.40	5.30
Neural Network	79.59	98.94	99.87

Choosing the best parameters to use as neural network inputs can lead to better results. Hamzeloo et al. (2014) combined the image segmentation approach presented in section 2.5.1 with neural networks to predict the PSD on the conveyor belt of the crushing circuit of a copper concentrator. Segmentation was first used to determine the contour of each particle. Ten features related to the particle size (e.g. particle area, diameter of maximum inscribed disk, dimensions of best-fit rectangle) were selected to predict the PSD. Then, by using Principal component analysis (PCA), the dimension of data was reduced to one. The selected feature was used as the input of an ANN. A multilayer perceptron neural network was used with one input, two hidden layers, including 6 and 3 neurons with one output. They used only 20 images to train and test the network and compared their results to sieve analyses. Although the number of images used for training and testing is minimal, and the input of the neural network consist in only one feature, the combination of segmentation with PCA-Neural network gave RMSE on the percentages passing between 7 and 25 %. As another example, Yaghoobi et al. (2019) utilized a neural network to find the PSD of fragmented rock. They extracted image features and utilized them as inputs for neural networks. Fourier transforms, Gabor filters, wavelet transforms, and their combinations were chosen as the image features in this. Results of manual segmentation achieved with commercial code Split were used as ground truth.

ANN are powerful tools, but they have limitations when working with types of data with large numbers of inputs, such as image, video or audio. As mentioned previously in this section, it is usually impossible to use images directly as ANN inputs. A type of neural network that can learn directly from images will be described in the next section.

2.5.4 Convolutional Neural Network (ConvNet)

ConvNets provide an ANN structure for inputs consisting of images, sound, and video (Abdel-Hamid, Jiang, and Penn, 2012; Chellapilla, Puri, and Simard, 2006; Gatys, Ecker, and Bethge, 2016; Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; LeCun and Bengio, 1995; Lee et al., 2017; Oquab, Bottou, Laptev, and Sivic, 2014; Yu et al., 2014). ConvNets apply convolutional filters to images or similar inputs to extract deep data for a fully connected neural network. A typical ConvNet uses millions of weight parameters in a large number of layers. It also requires thousands of images for training and parameter tuning (Lee et al., 2017). ConvNet are used in several fields, including medical image analysis (Hall et al., 2010; Lee et al., 2017; Plautz et al., 2017; Tajbakhsh et al., 2016; Yan, Lu, and Summers, 2018), traffic sign recognition (Sermanet and Lecun, 2011; Cireşan et al. 2012) and speech recognition (Abdel-Hamid, Deng, and Yu, 2013; Abdel-Hamid et al., 2012; LeCun and Bengio, 1995).

The most famous ConvNet was developed for image classification in 2012. Krizhevsky et al. (2012) won the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) competition to classify images of common objects. They got a winning top-5 test error rate of 15.3%, while the second error rate was 26.2%. Their ConvNet had 60 million parameters and 650,000 neurons. It consisted of five convolutional layers, followed by max pooling layers, and three fully-connected layers with a final 1000-way SoftMax. The architecture of their network, named AlexNet, was trained on the ImageNet dataset (Deng et al., 2009). This dataset contains over 15 million labelled high-resolution images in over 22000 categories.

Following Krizhevsky et al. (2012), other ConvNets were developed to reduce the ILSVRC prediction error. For example, Matthew Zeiler and Rob Fergus from New York University presented a model named ZFNet (Zeiler and Fergus, 2014) on ILSVRC 2013 and achieved an 11.2% error rate. Compared to AlexNet, ZFNet, used a smaller filter size in the first layer (7x7 compared to 11x11 in AlexNet). Also, ZFNet was trained on 1.3 million images, whereas AlexNet was trained on 15 million images. As another example, Simonyan and Zisserman (2014) presented a ConvNet named VGGNet in 2014 with an error rate of 7.3%. VGGNet, Visual Geometry Group in the Department of Engineering Science at Oxford University, uses a much simpler scheme. VGGNet further decreased the filter size to 3x3. They found that a stack of two conventional layers with a 3x3 filter has a 5x5 effective receptive field. As another model, GoogLeNet (Szegedy et al., 2015), a ConvNet model (with 22 weight layers), won the ILSVRC 2014 competition with a 6.7% error rate. GoogLeNet used a very deep model containing 100 channels as well as global average pooling instead of fully connected layers. This helps decrease the number of parameters. Finally, Microsoft ResNet (Wu et al., 2018,) a 152-layer network architecture, won ILSVRC 2015 with a 3.6% error rate. This error rate is less than the human error rate of 5-10%. Their idea on the residual block was to connect each layer to the next layer (like a traditional network) and to the layer with some layers skipping in between.

Determining the proper architecture is an important part of developing a ConvNet. The structure of ConvNet layers can be divided into three parts: convolutional filters, detection functions, and pooling functions. Convolutional filters are applied to the input (e.g. an image) to build a linear activation set. In the detection step, nonlinear activation functions like rectified linear activation function (ReLU), are applied to the linear activation. Pooling functions (e.g., max pooling) are frequently applied in the third part to replace the output of the network with summary statistics (like mean or maximum values) (Goodfellow, Bengio, and Courville, 2016).

Layers of feature maps are created by applying convolutional filters to the input image. Usually, low-level features, such as lines, arcs, and edges, are identified in the first convolutional layers (Lee, Kim, Kang, and Lim, 2018). Deeper convolutional layers generate deeper feature maps and can detect complex patterns in the images (Chollet, 2017). In the deeper layers of ConvNet, the size of the feature maps decreases but the number of feature layers increases (Figure 2.10). This can help find a larger number of details from each image in different feature layers.

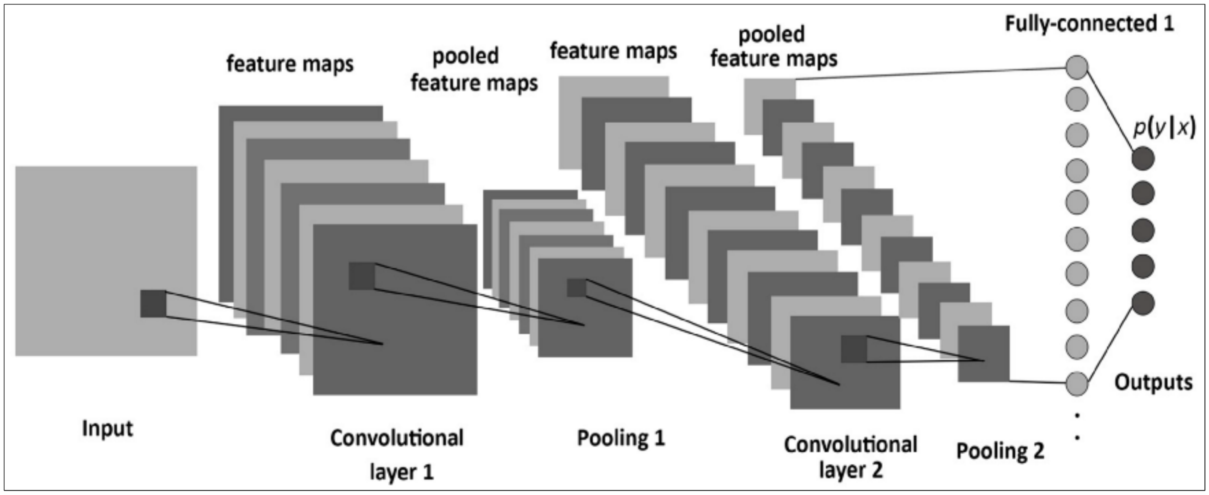

Figure 2.10 Example of ConvNet Structure. Taken from Blanc-Durand (2018)

The significant difference between ANN and ConvNet is that an ANN learns the overall pattern of all inputs, while a ConvNet finds local patterns and learns from them. There are two important features for ConvNets. First, they can learn a small pattern from one corner of the image and identify it throughout the image. This feature improves the efficiency of ConvNet while working with an image. Therefore, the learned patterns are translation invariant. The second important feature is that ConvNet can learn spatial hierarchies of patterns.

A convolutional layer consists in a sliding window of size *n*×*n* (normally n equals 3 or 5) with linear activation. The sliding window covers the input layer and extracts a feature for each window position with a convolution. The convolution operation consists in multiplying the linear activation obtained during training by the input values (Figure 2.11). The input layer is an image for the first layer or a feature map for the other layers. This sliding window is named convolutional kernel and is the key to finding the pattern with ConvNets. By applying m different kernels to each input layer, a feature map of depth m is created (Chollet, 2017).

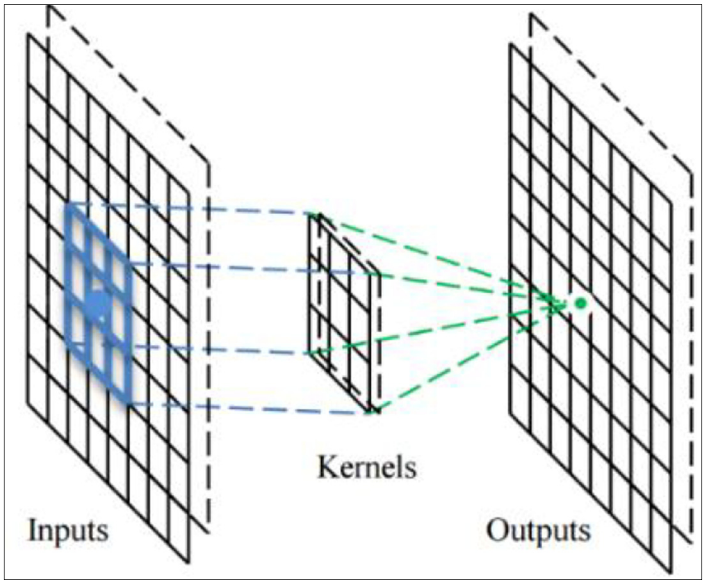

Figure 2.11 General principal of convolutional layer Yakura, Shinozaki, Nishimura, Oyama, and Sakuma (2018)

In the nonlinear layer, ReLU is used as an activation function:

$$f(z) = max\{0, z\} \tag{2.4}$$

where *f*(*z*) is the threshold function applied to *z* (Goodfellow et al., 2016). As shown in Figure 2.12, this function replaces by zero all negative numbers and does not alter the positive numbers. It typically increases the nonlinearity of the network.

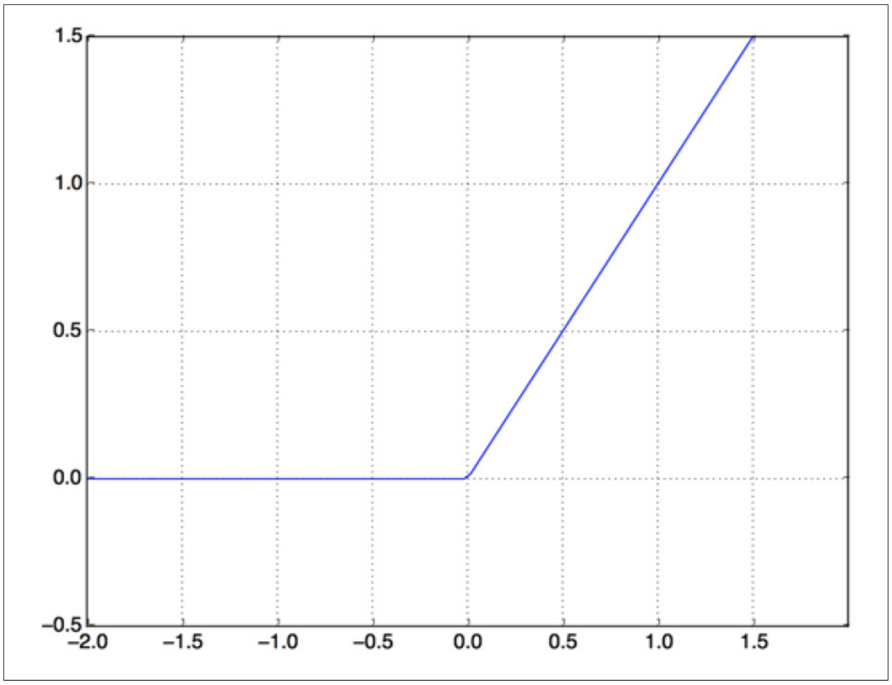

Figure 2.12 Rectified linear units (ReLU) function. Taken from Chollet (2017)

To reduce the size of the feature map and consequently reduce the number of computations, max and average pooling functions can be used (Figure 2.13). These functions down-sample the feature map using a pooling window which extracts the maximum or average value for each position of the window (Chollet, 2017). For example, a feature map of size 26×26 is converted to a feature map of size 13×13 by applying max pooling with a 2×2 window with a stride of 2 pixels. The main purpose of using pooling functions is to make the model invariant to input translation. For example, if there is a small translation in the input image, the results after applying pooling functions does not change too much (Goodfellow et al., 2016).

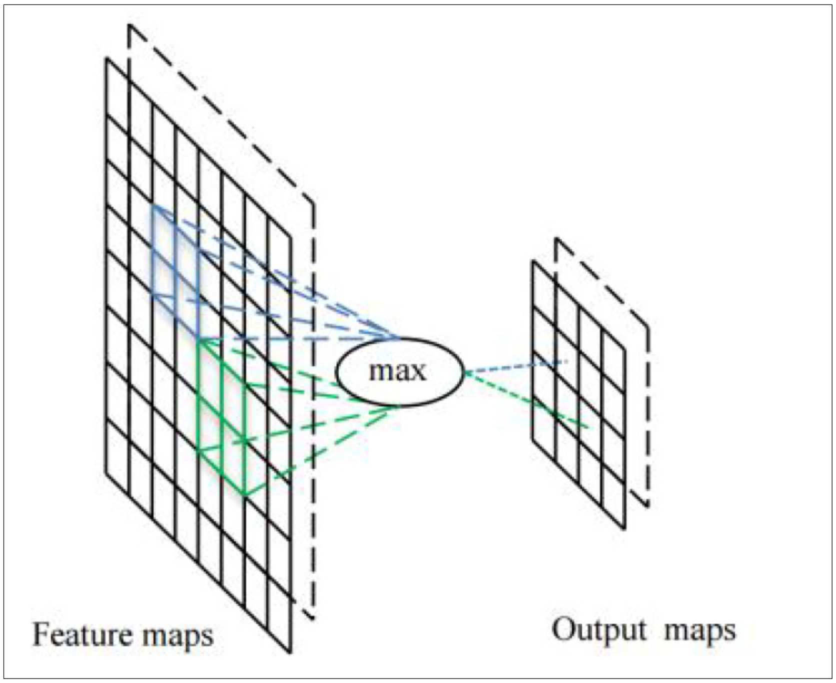

Figure 2.13 Visualization of a max-pooling function (size of 2×2 pixels, stride of 2 pixels) on a feature map. Taken from Lee et al. (2018)

ConvNets are increasingly common in image analysis. Still, a limited number of studies have examined particle sizing with ConvNets (Buscombe, 2020; Lang et al., 2021; McFall et al., 2020; Pirnia, Duhaime, and Manashti, 2018). This section presents four examples of networks that are using ConvNets to determine the size of bubbles or particles. The first two examples are centered on industrial applications. The other two examples describe ConvNet for PSD determination in sedimentology, an application closer to geotechnical engineering.

In food, pharmaceutical, and chemical industries, detecting bubbles and finding their size distribution is often essential. Ilonen et al. (2018) compared the accuracy of ConvNets for finding bubble size distribution with other methods, such as Boosting-based detection or Concentric Circular Arrangements Strokina. ConvNets performed better (Figure 2.14), but were slower. Also, the ConvNets cannot detect the size of each bubble but find instead the overall distribution.

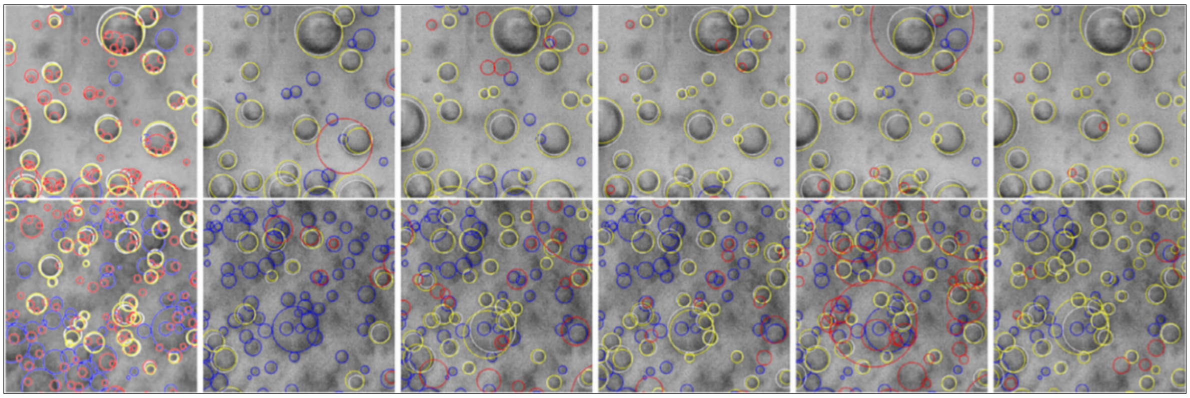

Figure 2.14 Bubble size detection results. ConvNet detected the right images (top and bottom rows), and others were detected by other methods. The ground truth markings are represented in white, true positives in yellow, false negatives in blue, and false positives in red. Taken from Ilonen et al. (2018)

Zhu et al. (2017) established a deep learning-based model, DeepEM, to recognize single particles from cryo-election microscopy. They used ConvNet as kernel of DeepEM to automate particle extraction from raw cryo-EM micrograph (Figure 2.15). The input image was 272×272 pixels, and three convolutional layers were applied before a fully connected layer.

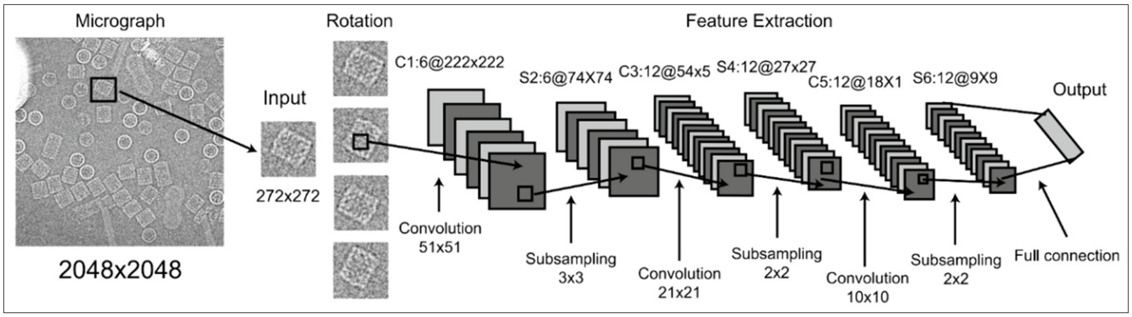

Figure 2.15 Schematics of ConvNet in DeepEM. Taken from Zhu et al. (2017)

GRAINet is one of the particle sizing ConvNets (Lang et al., 2021). It was created to determine the grain size distribution of gravel bars from images captured by unmanned aerial vehicles (UAV). Lang et al. utilized 1,419 photographs from 25 gravel bars and a variety of grain sizes ranging from about 0.5 cm to 40 cm collected from six rivers in Switzerland. GRAINet was fed 500×200-pixel images with normalized RGB channels. These images were obtained using a UAV (DJI PHANTOM 4 PRO) with a 20 megapixel camera. More than 180,000 particles were manually marked using human annotators. GRAINet comprises one single 3×3 "entry" ConvNet layer, followed by six residual blocks and a 1×1 ConvNet layer. Each of these residual blocks contained three convolutional layers. Each convolutional layer also encompassed one batch normalization and one ReLU layer. At the end of the network, the SoftMax function was used for grain size classification. The aggregate number of GRAINet parameters was about 1.6 million, which is relatively low compared to typical ConvNet models.

To evaluate the impact of image resolution on GRAINet, the original photos were resized into six different lower resolution images (Figure 2.16). GRAINet was then trained and run with the downsampled dataset. The mean absolute error (MAE) of the GRAINet diameter predictions increased to 1.4 and 1.9 cm when image size was decreased by factors of 16 and 40, respectively. MAE for the original image was around 1.1 cm.

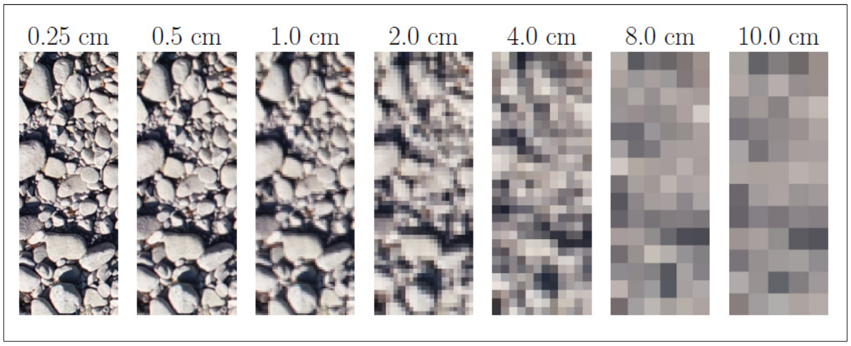

Figure 2.16 Original image with 0.25cm resolution at the left, followed by downgraded images with 0.5, 1.0, 2.0, 4.0, 8.0, 10.0 cm resolutions. Taken from Lang et al. (2021)

SediNet (Buscombe, 2020) is another example of ConvNets that predict the PSD and classify sediment images. SediNet was prepared for three regression estimates of nine grain size distribution percentiles. The two classification models included populations of grains and shape and size classes of grain. The image feature extractor of SediNet consisted of four convolutional blocks, each block containing convolutional layers, batch normalization layers, and max pooling layers. Subsequently, the batch normalization and max pooling layer and a dropout layer were added. Figure 2.17 shows a schematic representation of the SediNet architecture. The output of the feature extractor fed one of three blocks. The first block was used as a cumulative grain size predictor. Two classifier blocks were used to classify categorical parameters, including the population and shape of the grains.

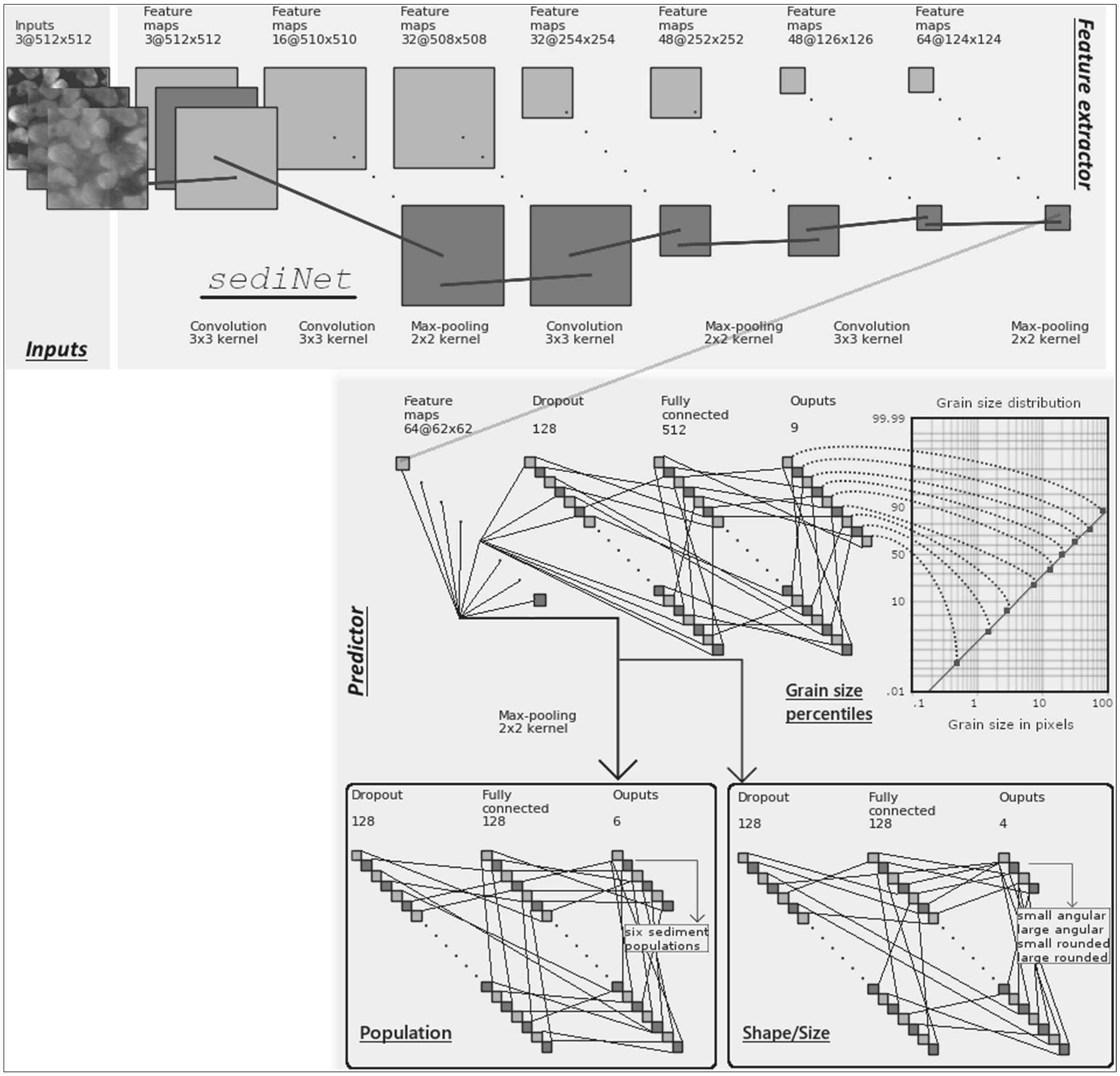

Figure 2.17 The schematic of the SediNet architecture. Taken from Buscombe (2020)

RGB images of 512×512 pixels were used as SediNet inputs. Nine cumulative grain size percentiles were predicted with the regression model. Subdatasets of 204 and 205 images were utilized to train and test SediNet, respectively. The mean training error ranged from 24 to 52%, while the mean testing errors varied from 24 to 45%. The second regression model was trained with 31 images of beach sand (16 images for testing and 15 images for training). For the training dataset, the mean error varied between 7% and 29%. For the test dataset, it varied between 16% and 29%. The third SediNet implementation learned and estimated the actual size of an image pixel. In this step, the training and testing subdatasets underestimated the grain size with a mean error of around 29% and 22%, respectively.

2.6 Granular material datasets

PSD determination based on photographs and machine learning requires databases of hundreds of images for neural networks and thousands of images for ConvNet. These images can be synthetic or real photographs (Nikolenko, 2021). This section presents some dataset examples and the datasets that were used in this project.

2.6.1 Synthetic images

The discrete element method (DEM) computes the acceleration, velocity and displacements of individual particles using Newton's second law and contact models that describe the forces between particles based on their overlap (Cundall and Strack 1979). The discrete element method is normally used to study the micromechanical behaviour of granular materials, but it can also be used to generate images.

YADE is an open-source DEM framework (Kozicki and Donzé, 2009; Šmilauer et al., 2015). It was written in C++ with a Python interface. It is a Linux-based program with the ability to simulate millions of particle interactions. YADE is used by researchers worldwide with an active community that has added many useful features, including particle-fluid coupling, interaction and grids, polyhedral particles, and FEM-coupling (Haustein, Gladkyy, and Schwarze, 2017). The YADE-open DEM framework is flexible when attaching new code for simulation and data from other codes can easily be imported. (Kozicki and Donzé, 2009).

Pirnia et al. (2018) developed a dataset of more than 53003 images using YADE. This dataset is used in the two papers presented in Chapters 3 and 4. A list of 53003 particle sizes between 75 and 1180 µm was created to cover all possible PSD for five sieves (106, 150, 250, 425, and 710 µm) and percentage passing increments of 5 %. The Java interface of COMSOL was utilized to control the python interface of YADE (Pirnia, Duhaime, Ethier, and Dubé, 2019). The particles were dropped into a transparent virtual box and allowed to settle. Views from the top and bottom of the transparent box were captured. Random colors were utilized to create the particles. Because each particle has a finite mass, the actual PSD can be slightly different from the defined PSD. However, the RMSE on the percentages passing is always lower than 0.2 %.

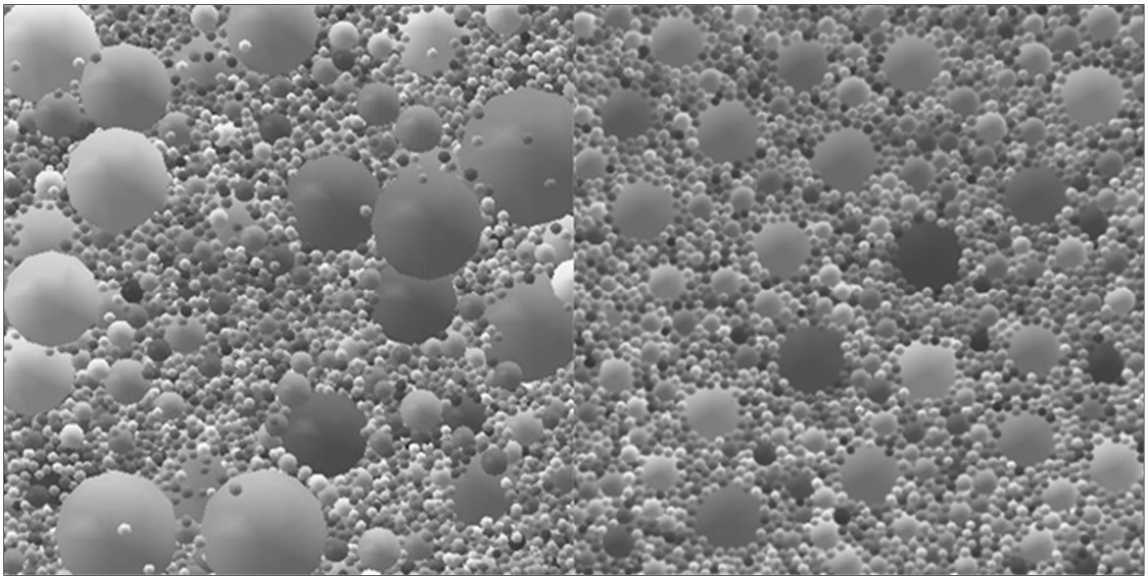

Figure 2.18 Right and left images show the bottom and top views of synthetic soil particles in the virtual transparent box

More realistic images were recently created with the Unity game engine by Temimi (2020). The Unity game engine is a cross-platform software created for creating 2D and 3D video games and interactive media (Haas, 2014). Unity is now used for a wide variety of usages like training, medical, education, architecture, physical installation and simulating (University of Queensland, 2016). Unity supports C# and UnityScript for programming to create and control interactive environments or special simulations (Unity Technologies, 2017). The Unity game engine can be used to simulate particles like sand and rocks. Temimi (2020) prepared a dataset of 3003 images varying between 75 and 1180 µm (Figure 2.19). The images contained thousands of particles with different sizes and shapes concerning a series of imposed grain sizes. This method could expand the project database with a limited number of images to provide better results. Two textural feature extraction methods, Haar wavelet transforms, and Haralick features, were used to predict the PSD for this dataset. The RMSE value for the percentages passing was 8.4% on average.

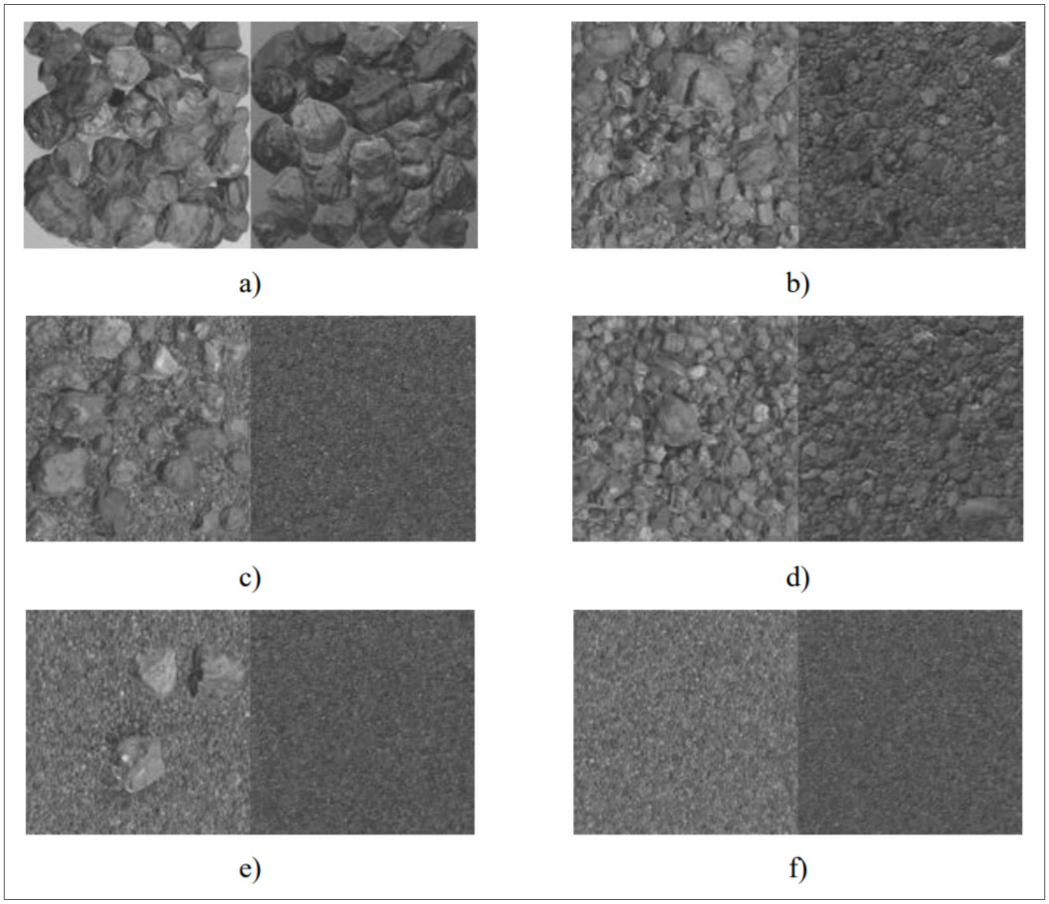

Figure 2.19 Sample of top and bottom views of more realistic images created by Temimi (2020) for coarse particles (a and b), relatively widely-graded particles (c and d), and fine particles (e and f)

2.6.2 Real soil photographs

Most research on particle sizing is based on real image datasets. For instance, SediNet (Buscombe, 2020) was trained and evaluated with 409 real soil images. It is important to know that the training and testing of ConvNet models require thousands of images to avoid overfitting (Goodfellow et al., 2016). For this reason, Lang et al. (2021) utilized a larger dataset for training and testing GRAINet including 1,419 images. Some of these images were obtained by flipping images from a smaller dataset horizontally and vertically. Likewise, Mcfall et al. (2020) divided and flipped 63 images to provide 517 beach sand images for training and testing SediNet.

The feature extraction and ConvNet methods developed in Chapters 3 and 4 with synthetic images needed to be tested using a dataset composed of thousands of real photographs. The materials prepared and photographed by Telcy (2018) were used. The preparation of this dataset by Telcy (2018) is described below.

Three soil types (Bomix sand, 2C sand, and Till) were selected. To increase the number of materials, the Bomix and 2C sands were separated into retained and passing fractions using sieves of 425 and 625 μm, respectively. Consequently, five granular materials were prepared: Retained Bomix, Passing Bomix, Retained 2C, Passing 2C, and Till.

The total mass of each granular material (20 kg) was first divided in two using a riffle splitter. For each material, a subsample of 10 kg was divided into eight subsamples of 1.25 kg using a rotary soil splitter (spinning riffler) (Figure 2.20). The PSD of the five materials was determined using one of the eight subsamples according to standard ASTM D6913. Six of the other subsamples were combined to prepare 15 samples of 2.5 kg. The 15 materials correspond to the five original materials and each possible 1:1 mixture. The real PSD of the mixtures was determined from the PSD and the mass of their components.

A total of 900 photographs were taken for the 15 materials by Telcy (2018). Each material was poured in a pan. The images were taken from the top using a Nikon D90 with ISO 400 sensitivity and a focal length of 55 mm. For each material three series of 20 photographs were taken with different distances between the camera and sample. The material was mixed in the pan between each photograph.

A set of 300 images corresponding to the same camera-sample distance was used for the third paper. A dataset of this size is sufficient to train a neural network with traditional features (Manashti. Duhaime, Toews, Pirnia, 2021), but too small for training a ConvNet. For creating a dataset containing 9600 photographs, each image was split into 32 subimages of 512×512 pixels. This dataset is suitable for training, verifying, and testing the ConvNet model. On the other hand, the image splitting is a source of error as the PSD of each subimage is different because of segregation (Thurley and Ng, 2008). The subimages located in the original image's corners also tended to be slightly out of focus and blurry. As seen in Chapter 5, the results of the particle size analysis with PSDNet for this dataset show that these errors do not significantly influence the results.

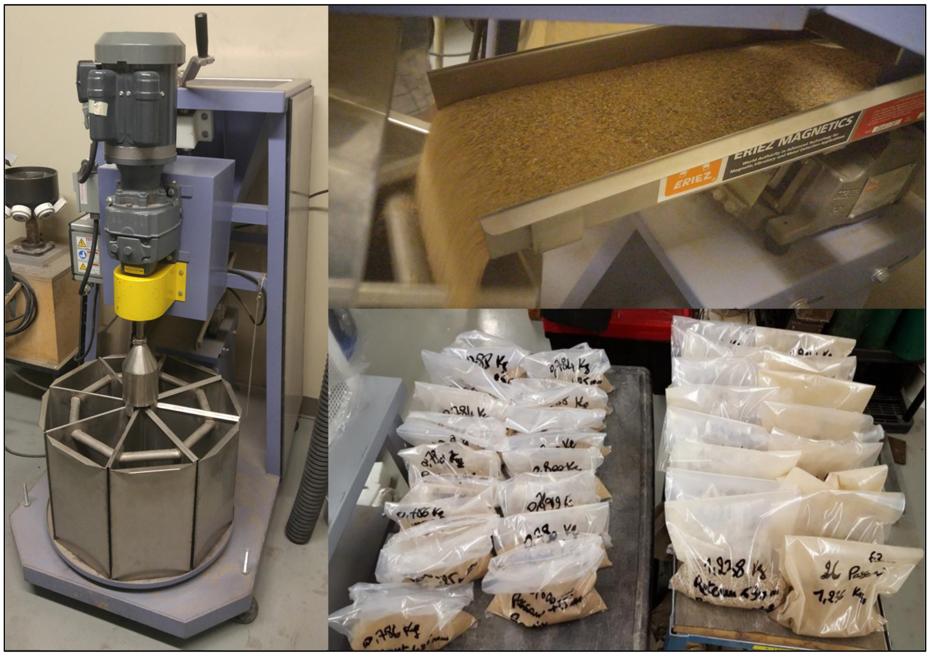

Figure 2.20 Rotary Soil Splitter to prepare subsamples

It is important in geotechnical engineering to use sieving as the ground truth for particle sizing. Segmentation was selected as ground truth by Buscombe (2020) and McFall et al. (2020). Sieving was used to do particle sizing of the real soil samples (ground truth) in this research. On the other hand, creating and improving a ConvNet model requires thousands of images. The limited number of images in the training, validation, and test dataset, is the most significant weakness of GRAINet (Lang et al., 2021) with 1,419 pictures and SediNet (Buscombe, 2020) with 409 images.

CHAPTER 3

Grain Size Analysis of Spherical Particles Based on Image Feature Extraction

Javad Manashti[1, 3], François Duhaime[1], Matthew 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, Montreal, Canada*

[3] *Corresponding author, Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, 1100 Notre-Dame Ouest, Montreal, Quebec, H3C 1K3, Canada*

Paper submitted for publication, December 2021

3.1 Abstract

The objectives of this paper were to demonstrate how the discrete element method can be used to generate images of particle assemblages and to compare the performance of different feature extraction methods to determine the particle size distribution (PSD) of spherical particles from 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, Histogram of Oriented Gradients, Entropy, Local Binary Pattern, Local Configuration Pattern, Complete Local Binary Pattern (CLBP), Fast Fourier transform, Gabor filter, and Haar Discrete Wavelet Transform. The feature extraction methods were used to generate the inputs for 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

3.2 Introduction

Sieving, the current standard method for PSD determination in geotechnical engineering (ASTM C136/C136M, 2019a), dates back from 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 (Bardet, 1997). 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 (ASTM D2487, 2017). 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 (Tuceryan and Jain, 1993).

Commercial codes such as WipFrag (Maerz et al., 1996), and VisioRock (Guyot, Monredon, LaRosa, and Broussaud, 2004) 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 (Chavez, Cheimanoff, and Schleifer, 1996; Thurley, 2011). 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 (Thurley, 2011).

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. (2006) compared automatic segmentation using commercial codes WipFrag and Fragalyst (Raina, Choudhury, Ramulu, Chakraborty, and Misra, 2002), 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 (1996) compared the PSD obtained with commercial segmentation codes FragScan (Cheimanoff, Chavez, and Schleifer, 1993), WipFrag (Maerz et al., 1996) and Split (Girdner, Kemeny, Srikant, and McGill, 1996), 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. (2006) and Liu and Tran (1996) 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 (Liu and Tran, 1996). When comparing only the segmentation, RMSE of 15 and 20 % were obtained for WipFrag and Fragalyst (Sudhakar et al., 2006).

Textural features are functions that describe the spatial arrangement of pixel intensities (Liu et al., 2019; Tuceryan and Jain, 1993). Methods for the extraction of textural features can be divided in three main categories (Tian, 2013). Methods of the first type use statistical parameters to describe the spatial distribution of pixel intensities. For example, the Haralick features (Haralick, Shanmugam, and Dinstein, 1973) 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) (Ojala, Pietikainen, and Maenpaa, 2002)). The third type includes transform-based feature extraction methods, such as Gabor wavelets (Manjunathi and Ma, 1996).

Statistical textural parameters have been used for PSD determination or rock classification. Itoh et al. (2008) compared the performance of 32 textural and color 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 (Ojala et al., 2002). 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 (Ojala et al., 2002). These methods are generally used for classification purposes. For example, Paclík et al. (2005) 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 (Hryciw et al., 2015; Shin and Hryciw, 2004). 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 (2004) 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 color. However, the normalization of the wavelet energy was shown to eliminate the effect of illumination and particle color. Yaghoobi et al. (2019) 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 (Ghalib, Hryciw, and Shin, 1998; Raza Ansari, 2018; Saghatoleslam et al., 2004). 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. (1998) 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 (2011a) 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. (2014) 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. (2019) 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 particles assemblages 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.

3.3 Methodology

3.3.1 Synthetic granular image dataset

A large image dataset was prepared by Pirnia et al. (2018) 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 (Pirnia et al., 2019). 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 (Figure 3.1). 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 %.

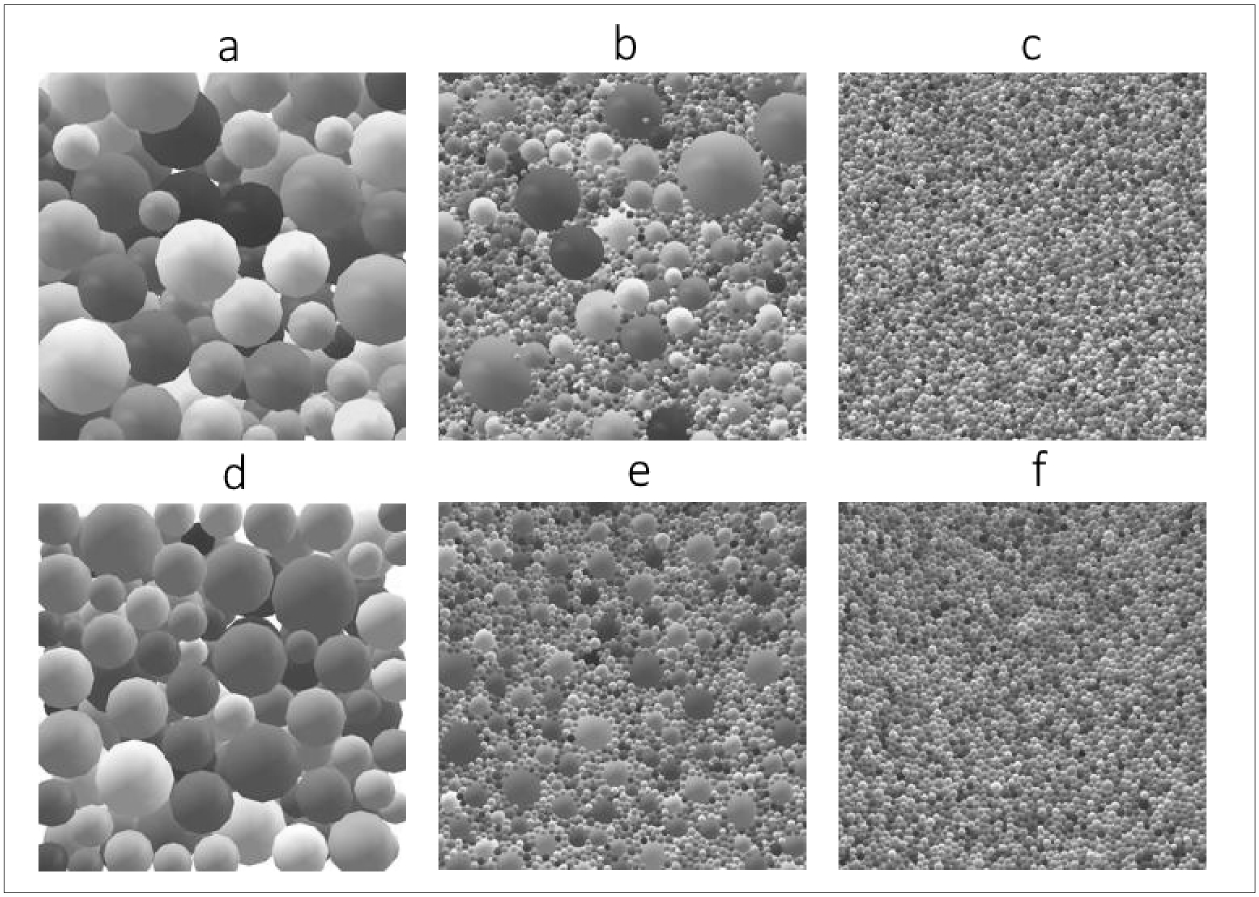

Figure 3.1. Top (upper row, a-c) and bottom (lower row, d-f) aaviews of the synthetic particles in the virtual transparent box. Left (a, d), center (b, e) and right (c, f) images show examples of coarse, widely-graded, and fine PSD

3.3.2 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.

3.3.2.1 Haralick Features

Haralick features are statistics of the gray level cooccurrence matrix (GLCM) (Gadkari, 2004; Haralick et al., 1973; Itoh et al., 2008). 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. (1973) were calculated for each GLCM (total of 56 features) using the MATLAB script of Monzel (2019). Preliminary tests showed that some Haralick features are better correlated with the percentage passing. This is the case of correlation and contrast:

$$Correlation = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} i\ j\ GLCM(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y} \qquad (3.1)$$

$$Contrast = \sum_{n=0}^{N_g-1} n^2 \underset{|i-j|=n}{\sum_{j=1}^{N_g}\sum_{i=1}^{N_g} GLCM(i,j)} \qquad (3.2)$$

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

$$p_x(i) = \sum_{j=1}^{N_g} GLCM(i,j) \qquad (3.3)$$

$$p_y(j) = \sum_{i=1}^{N_g} GLCM(i,j) \qquad (3.4)$$

Correlation describes the linear dependency in pixel intensity and contrast increases when the image contains abrupt changes in pixel intensity (Gadkari, 2004).

3.3.2.2 Histogram of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) describe the image by counting the occurrence of different orientation of the gray intensity gradient for a series of predefined cells (Dalal and Triggs, 2005). 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 (Pang, Yuan, Li, and Pan, 2011). Figure 3.2 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.

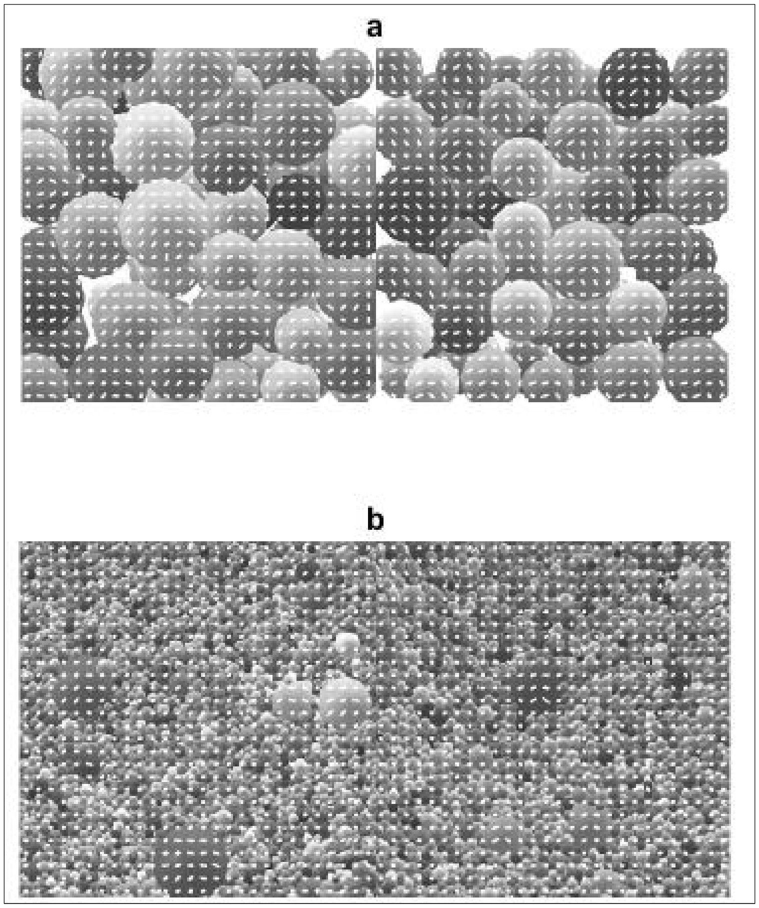

Figure 3.2. Examples of gradient orientations calculated for a cell size of 8 pixels for a) coarse and b) fine PSD

In this study, square HOG cells with sizes of 2, 4, 8, 16, 32, and 64 pixels were used. MATLAB function extractHOGFeatures (MATLAB, 2019) 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.

3.3.2.3 Local entropy of grayscale image

Entropy (Gonzalez and Woods, 2018) 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:

$$H(X) = -\sum_{j=1}^{N_g} P_j \; log_2 P_j \tag{3.5}$$

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 (Larkin, 2016).

The neighborhood around the center pixel is defined using a filter. Figure 3.3 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. Figure 3.4 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.

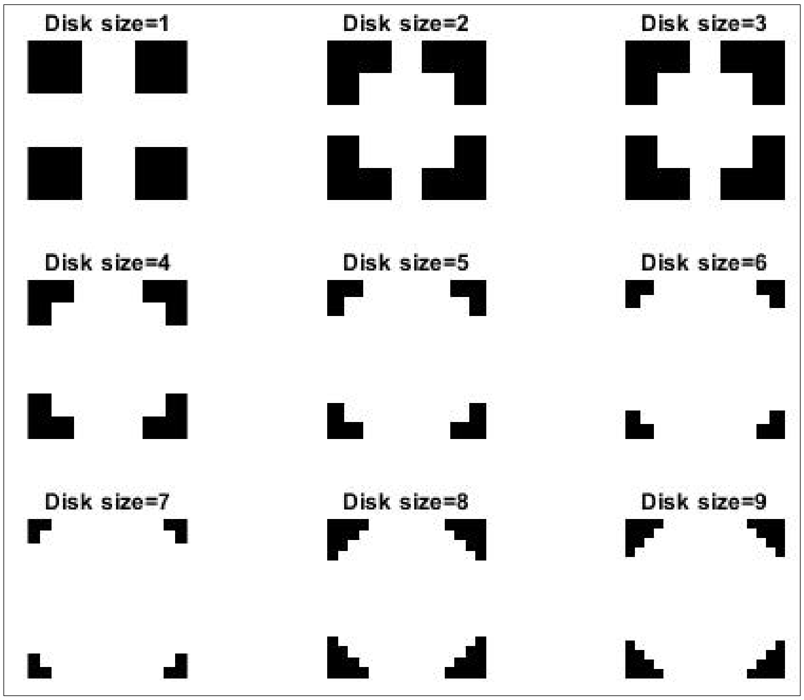

Figure 3.3. Nine disk filter sizes used to calculate entropy

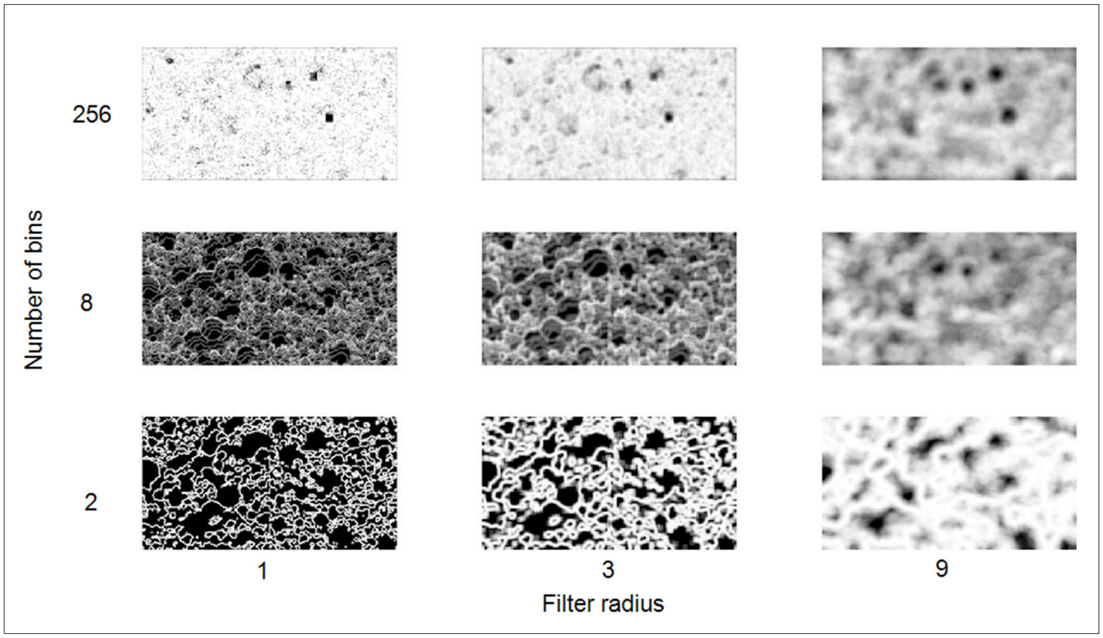

Figure 3.4. Effect of disk filter size and number of bins on the local entropy for the widely graded PSD (Figure 3.1b and 1e). Coarser particles are associated with lower values of the local entropy (darker gray levels)

3.3.2.4 Local Binary Pattern (LBP)

The Local Binary Pattern (LBP) sets the center pixel of a neighborhood as a threshold (Ojala et al., 2002). Neighboring pixels take a value of 0 or 1 based on the gray level difference between them and the center pixel (Sudha and Ramakrishna, 2017). 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:

$$LBP = \sum_{p=0}^{p-1} s\left(x_p - x_c\right) 2^p \tag{3.6}$$

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:

$$s\left(x_p - x_c\right) = \begin{cases} 1, x_p - x_c \geq 0 \\ 0, x_p - x_c < 0 \end{cases} \tag{3.7}$$

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 (Kaya, Ertuğrul, and Tekin, 2015). 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.

3.3.2.5 Local configuration pattern (LCP)

The Local Configuration Pattern (LCP) combines LBP with microscopic configurations (MiC) (L. Deng and Yu, 2015; Guo, Zhao, and Pietikäinen, 2011). 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:

$$\begin{bmatrix} x_{c,0} \\ \vdots \\ x_{c,N} \end{bmatrix} = \begin{bmatrix} x_{0,0} & \cdots & x_{p,0} \\ \vdots & \ddots & \vdots \\ x_{p,N} & \cdots & x_{p,N} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_p \end{bmatrix} \tag{3.8}$$

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. 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.

3.3.2.6 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 (Guo, Zhang, and Zhang, 2010). 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 ($|\mathrm{x_p} - \mathrm{x_c}|$). 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(Ojala et al., 2002), LCP(Y. Guo et al., 2011), and CLBP(Guo, Zhao, and Pietikäinen, 2012) were obtained from the Center for Machine Vision Research, Department of Computer Science and Engineering, University of

Oulu, Finland (*Center for Machine Vision Research, Department of Computer Science and Engineering, Matlab Code*, n.d.).

3.3.2.7 Fourier Transform

The two-dimensional discrete Fourier transform $F(u, v)$ of an image with pixel intensity f(*x*, *y*) is defined as follows (Szeliski, 2011; Yaghoobi et al., 2019):

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-2\pi\sqrt{-1}\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad (3.9)$$

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 2D FFT magnitude and its standard deviation was used to find the PSD.

Figure 3.5 shows the magnitude spectrum of the Fourier Transform for the images shown in Figure 3.1a, 3.1b and 3.1c. 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 Figure 3.5a is associated with coarser particles in the original image. This brighter center can be compared with the darker center in Figure 3.5c associated with fine particles.

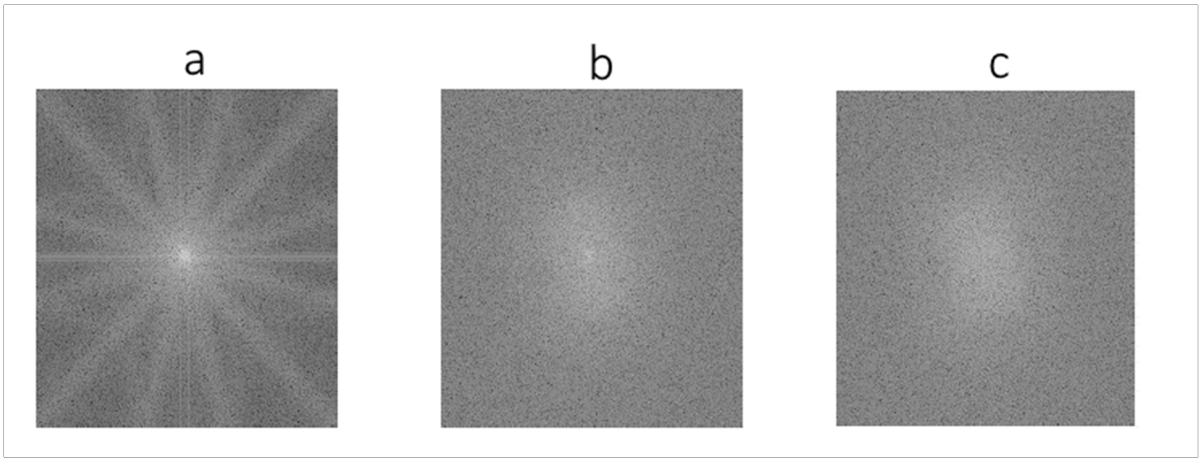

Figure 3.5. Magnitude spectrum of the Fourier transform for images corresponding to the coarse (a), widely-graded (b), and fine (c) PSD in Figures 1a, 1b, and 1c, respectively

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 (Yaghoobi et al., 2019). 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 Figure 3.6 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 Figure 3.6 shows eight rings with a constant 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.

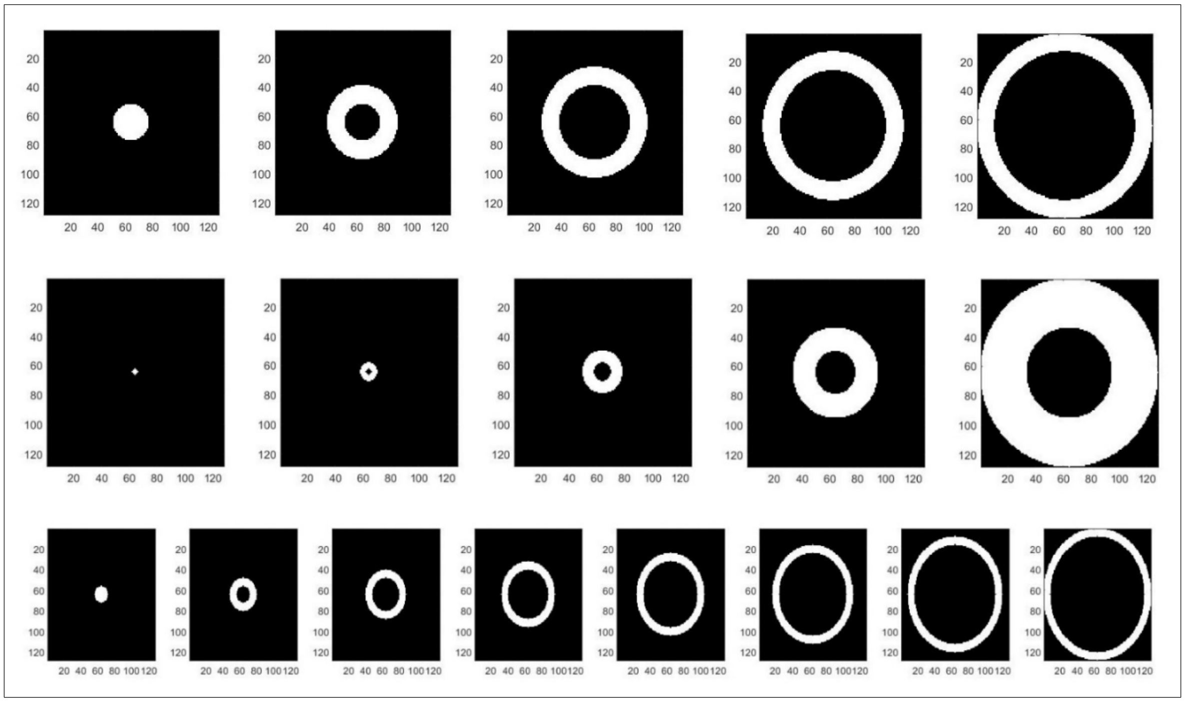

Figure 3.6. Examples of concentric circular filters for which the mean and standard deviation of the Fourier transform magnitude spectrum is calculated. The first row shows five filters with equal thickness (X=1.0). The second row illustrates five filters in which the thickness of every ring is twice the previous ring thickness (X=2.0). The third row shows eight filters with an equal thickness

3.3.2.8 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 (Tuceryan and Jain, 1993; Yaghoobi et al., 2019). 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:

$$F(x,y) = \left(\frac{1}{2\pi S_x S_y}\right) exp\left(-\frac{1}{2}\left(\frac{X^2}{{S_x}^2} + \frac{Y^2}{{S_y}^2}\right)\right) cos(2\pi u_0 X) \tag{3.10}$$

$$F(u,v) = \frac{1}{2}\left\{exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)}{S_u^2} + \frac{v^2}{S_v^2}\right]\right\} + exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)}{S_u^2} + \frac{v^2}{S_v^2}\right]\right\}\right\} \tag{3.11}$$

$$S_u = \frac{1}{2\pi S_x}, S_v = \frac{1}{2\pi S_y} \tag{3.12}$$

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.

Figure 3.7 shows the Gabor filter magnitude for a wavelength of 3,9, and 15 pixels and an 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.

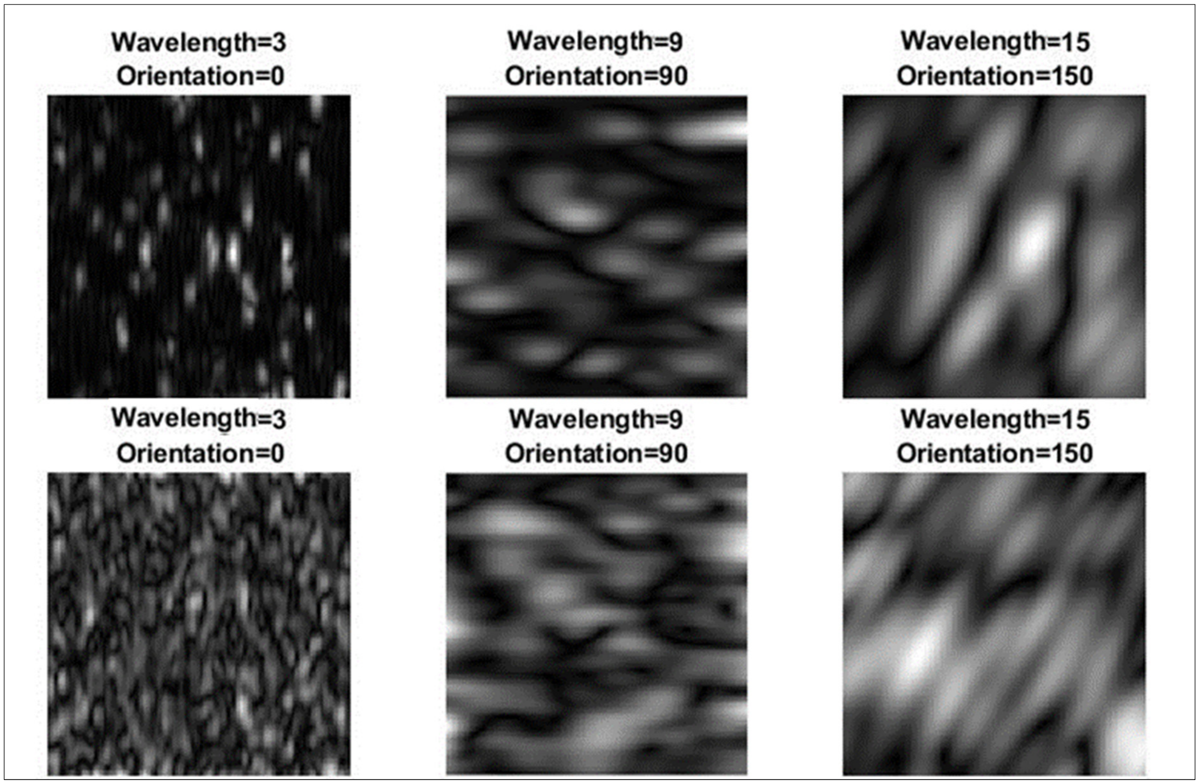

Figure 3.7. Magnitude of 3 selected Gabor filters with different wavelengths and orientations for the coarse (top row) and fine (second row) PSD

3.3.2.9 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 (Gonzalez and Woods, 2018):

$$\psi_{u,v} = \begin{cases} \sqrt{u} & 0 \leq ux - v \leq 0.5 \\ -\sqrt{u} & 0.5 \leq ux - v \leq 1.0 \\ 0 & elsewhere \end{cases} \tag{3.13}$$

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 (Gonzalez and Woods, 2018):

$$I = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{3.14}$$

Applying the transform results in the following four coefficients:

$$A = \frac{a+b+c+d}{2} \tag{3.15}$$

$$H = \frac{a+b-c-d}{2} \tag{3.16}$$

$$V = \frac{a-b+c-d}{2} \tag{3.17}$$

$$D = \frac{-a+b+c-d}{2} \tag{3.18}$$

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 (Ohm and Hryciw, 2014b). 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. Figure 3.8 presents one example of the successive application of Haar wavelet transforms and the resulting approximations.

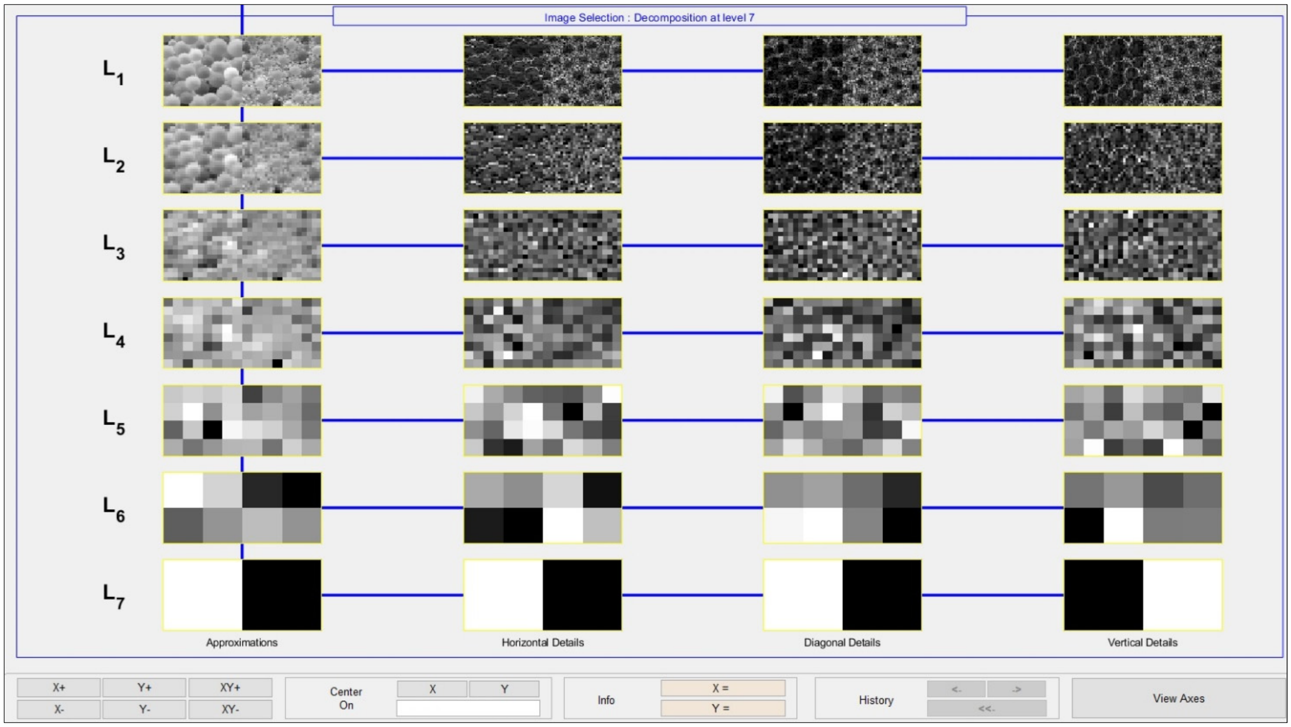

Figure 3.8. Seven levels of wavelet approximations for horizontal, diagonal, and vertical orientations

3.3.3 Artificial Neural Network

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 (Figure 3.9). 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.

For training of the neural networks, 70% of the 53 003 images were used for training, 15% for validation, and 15% for testing (Chakraborty et al., 2019; Gholamy et al., 2018). 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 technique was utilized for the network training. Figure 3.10 illustrates the performance of the neural network during training for a selection of the best features for all methods. This selection set contains 618 features, including 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).The RMSE decreases for the training, test, and validation datasets during the network training until the maximum number of epochs is reached (5000).

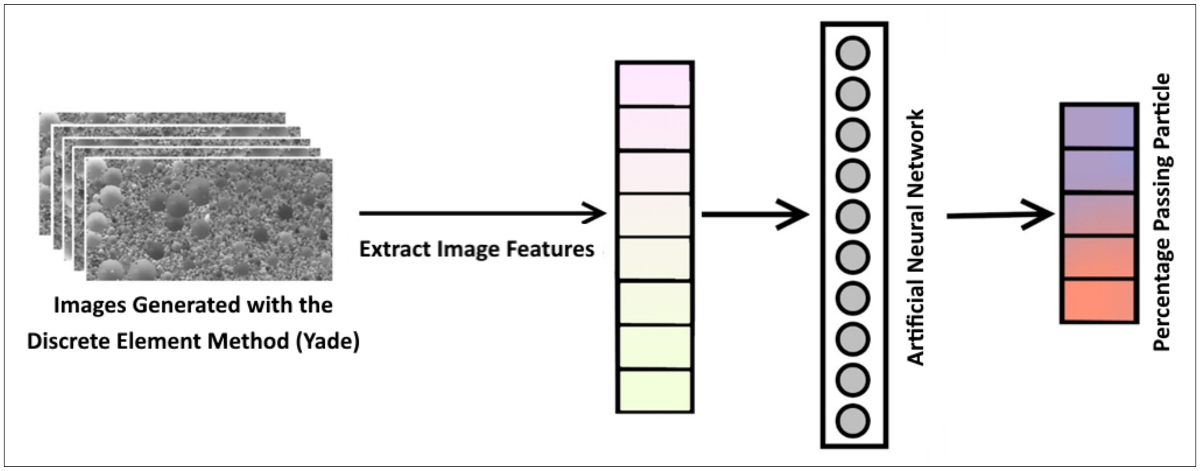
Images Generated with the
Discrete Element Method (Yade)
Extract Image Features
Artificial Neural Network
Percentage Passing Particle

Figure 3.9. Artificial Neural Network to predict the percentage passing for a series of sieves

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 (Krizhevsky et al., 2012; L. Liu et al., 2019).

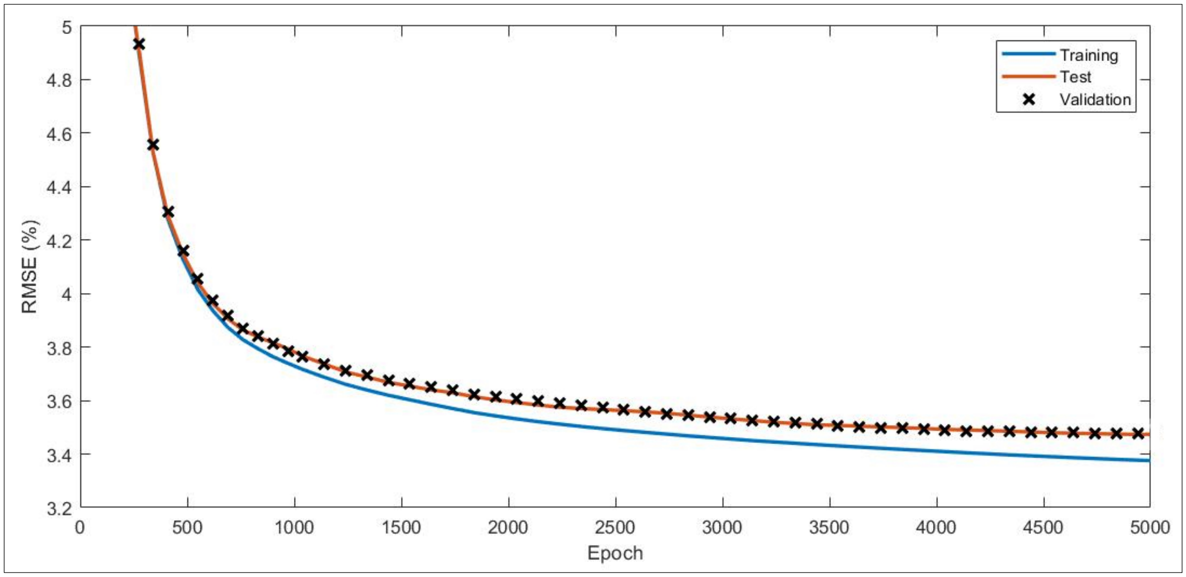

Figure 3.10. Neural network training performance for a selection of the 618 best features

3.4 Results

3.4.1 Entropy

Different number of bins were used to calculate the local entropy. Figure 3.11 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, the lowest RMSE (3.5 %) was obtained with 4 bins for the finest sieve. Except for 256 bins, better performances were obtained for the finest sieve. This was also observed for the other types of features.

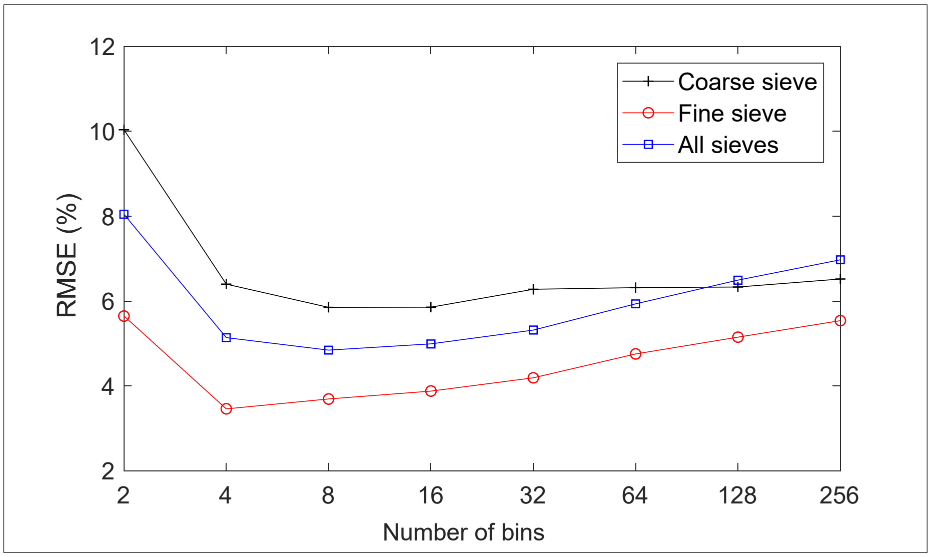

Figure 3.11. RMSE on the percentage passing for the fine sieve, coarse sieve, and all sieves using local entropy with different bin sizes using the test dataset

3.4.2 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. Figure 3.12 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 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.

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 decreases 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 could 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).

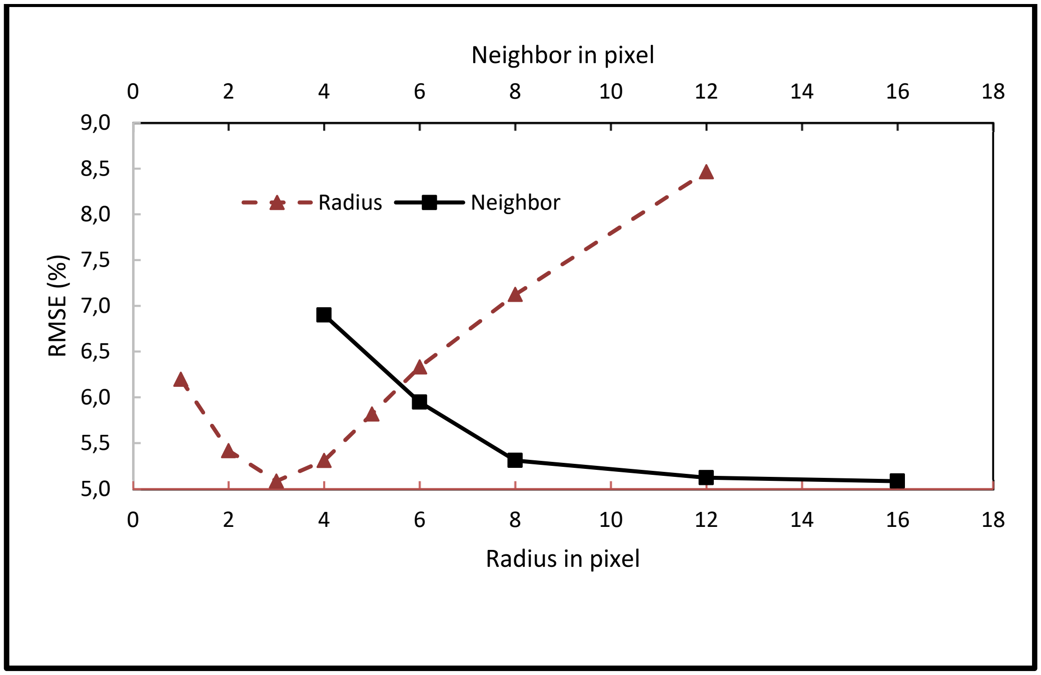

Figure 3.12. RMSE on the percentage passing for LBP with different radius values and number of neighbor pixels for all sieve sizes

As with LBP, the RMSE for LCP and CLBP decreased with an increasing number of neighbors. Figure 3.13 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.

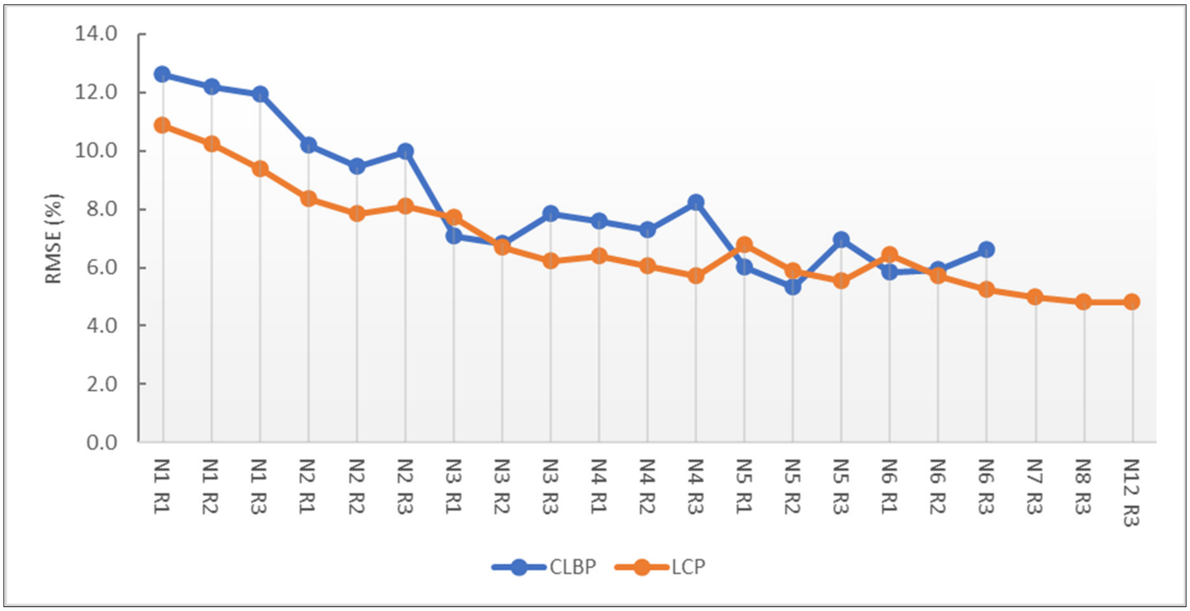

Figure 3.13. RMSE of LCP and CLBP for different number of neighbors and radius for all sieve sizes. N refers to the number of neighbors and R to the radius

3.4.3 Fourier Transform

Figure 3.14 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 radius 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.

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. (2019) who used 20 equal rings to predict the size of fragmented rock, does not significantly increase the accuracy of the PSD determination.

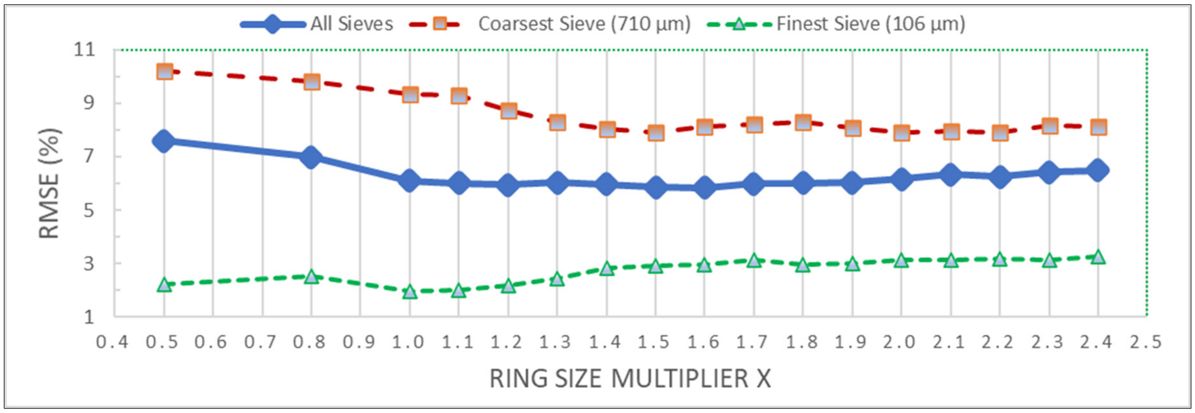

Figure 3.14. RMSE results for the mean and standard deviation of the Fourier transform magnitude spectrum of 5 rings with size multipliers from 0.5 to 2.4

3.4.4 Other methods and combination of all features

Figure 3.15 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).

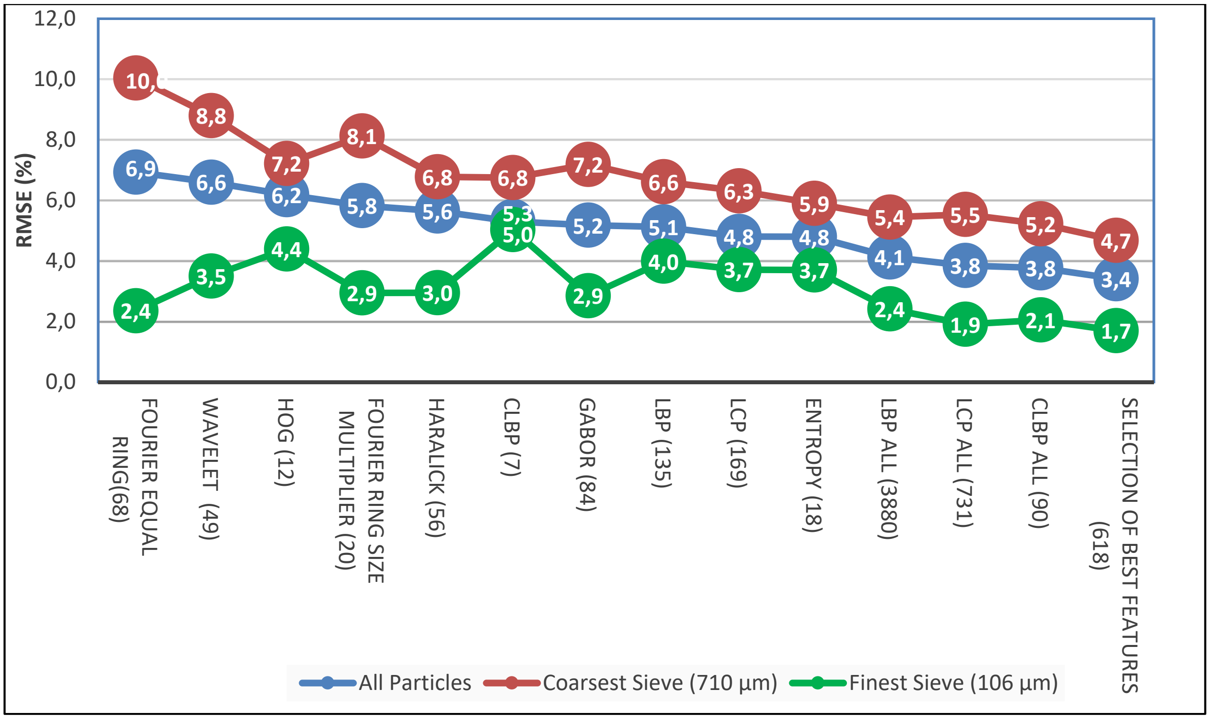

Figure 3.15. Comparison of the RMSE on the percentage passing for each type of extracted features and some selected combinations. The number of features for each method is given between parentheses

Irrespective of the type of textural feature and sieve size, the RMSE of 1.7 to 10 % in Figure 3.15 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 (1996) 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 (Sudhakar et al., 2006). On the other hand, it should be realized that the synthetic soil photographs (Figure 3.1) are idealized and probably much easier to handle than real soil photographs with varying illumination, particle color, and particle shape.

Figure 3.16 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.

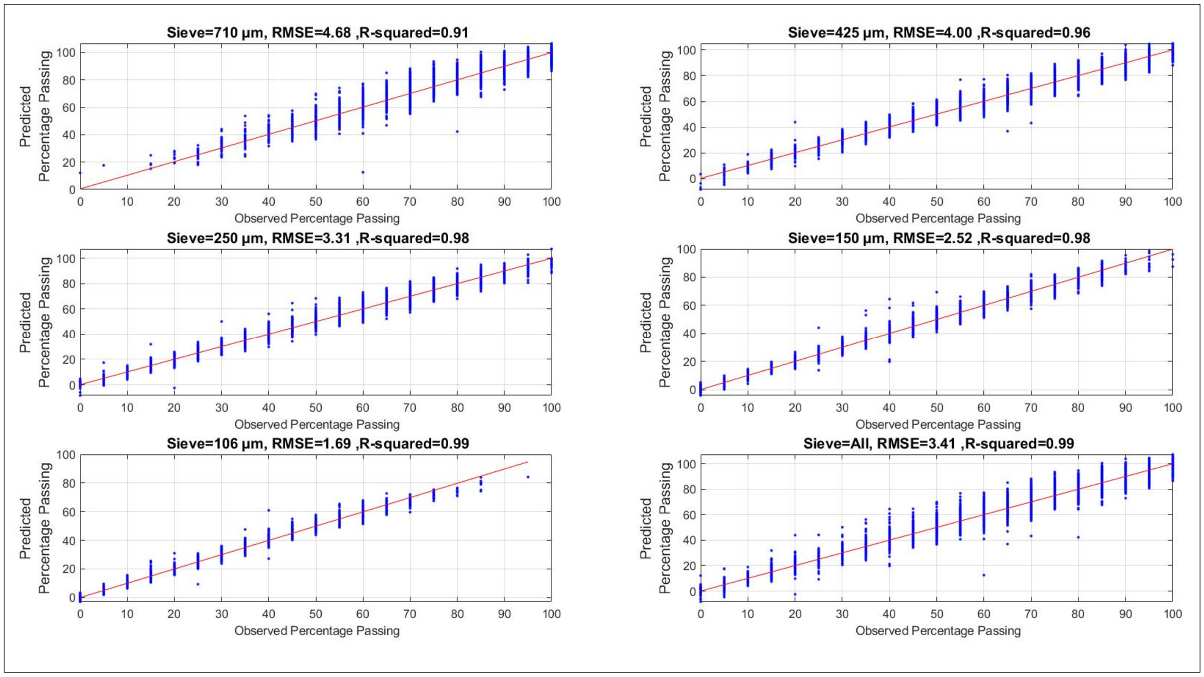

Figure 3.16. Comparison of the real and predicted percentages passing for a selection of the 618 best features

Figure 3.17 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 varies between 1.4 and 5.2 %. Overall, the real and predicted PSD are similar, even for a RMSE of 5.2%.

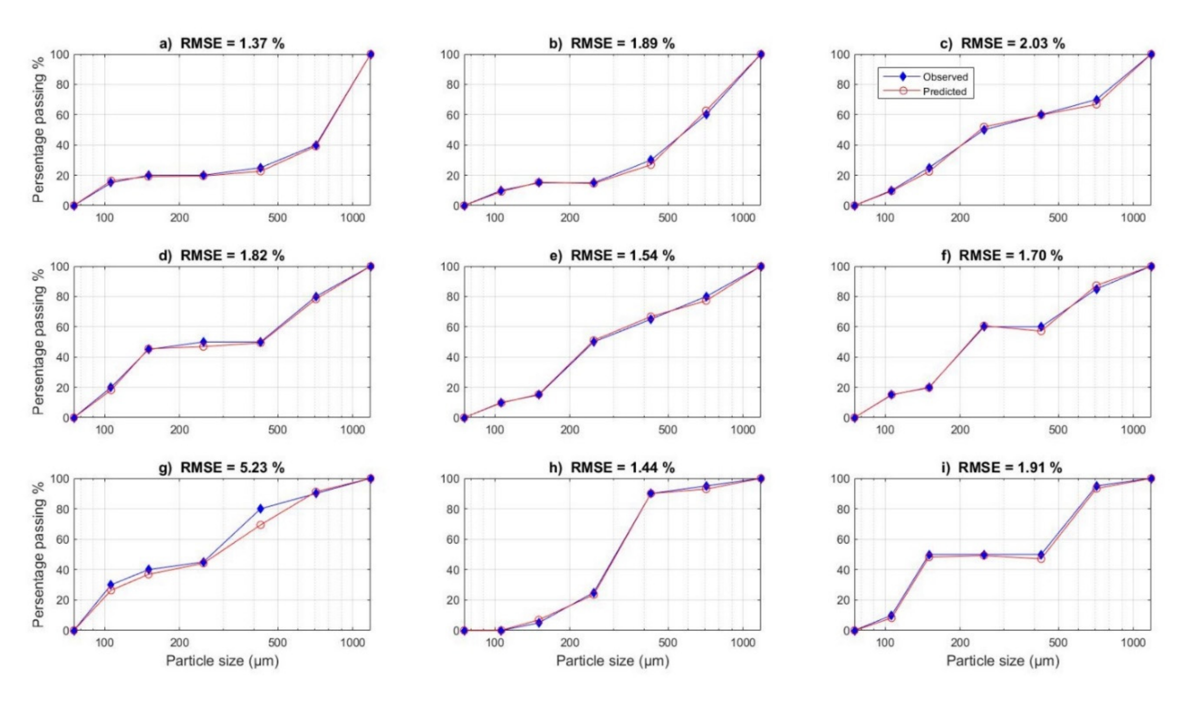

Figure 3.17. Examples of real and predicted PSD for the synthetic images obtained with the DEM and the 618 best features

The error percentage on *D₅₀* was calculated for the combination of 618 features. The *D₅₀* for both the predicted and observed PSD were calculated based on the method presented by Thyabat et al. (2007). The *D₅₀* error for the combination of all methods is 6.1 %. The predicted and observed *D₅₀* values are compared in Figure 3.18. As with the RMSE on the percentages passing, the error on the *D₅₀* is low compared to the values presented in the literature for real photographs. For example, Liu and Tran (1996) obtained *D₅₀* errors of approximately 55 % with Split and WipFrag. Again, the good performances for this dataset were expected considering its idealized nature.

In some cases, the predicted *D₅₀* was not accurate because of the shape of the PSD. An example of such PSD is shown in Figure 3.17i. 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 *D₅₀* undefined. These cases correspond to the vertical lines in Figure 3.18 at observed *D*$_{50}$ values corresponding to sieves sizes of 150, 250 and 425 µm.

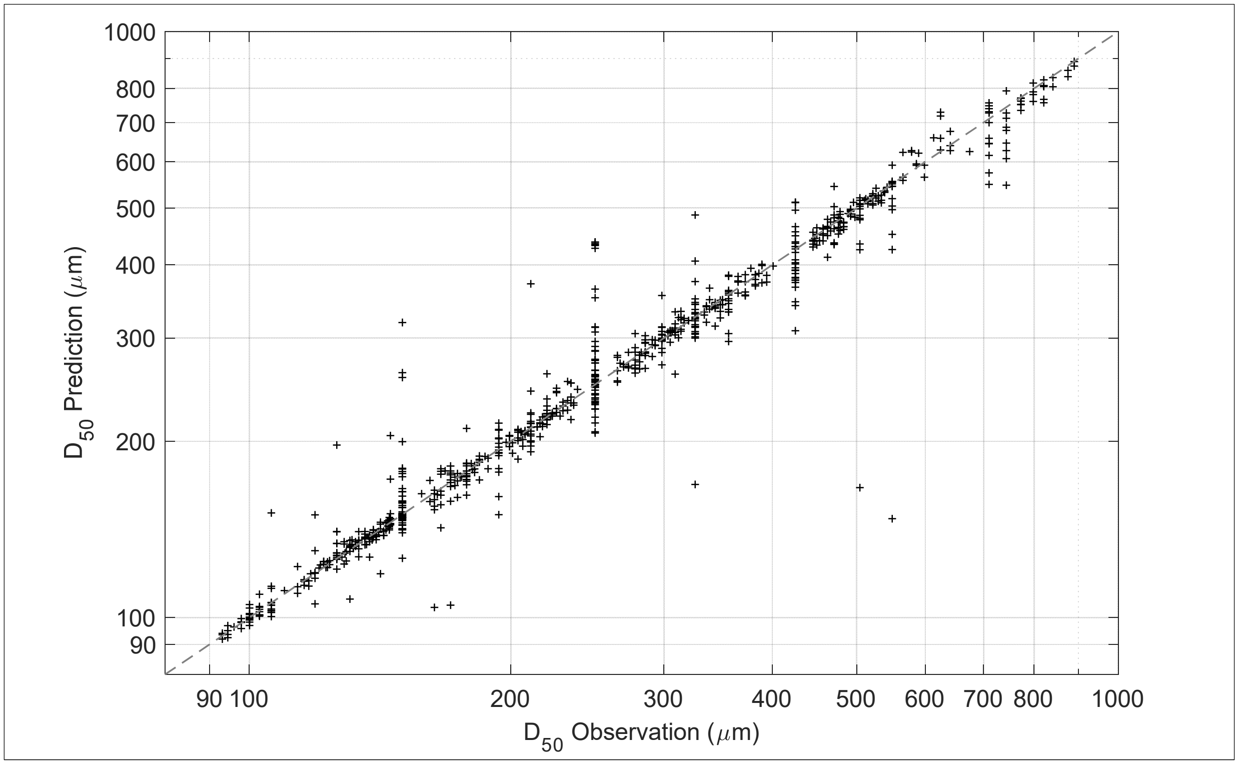

Figure 3.18. Prediction and Observation of D50 for the 618 best features

3.5 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, 2017), 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.

3.6 Data Availability

Codes for feature extraction are available from the corresponding author upon reasonable request. The image dataset is available from Pouyan Pirnia (pouyan.pirnia@gmail.com) upon reasonable request.

3.7 Acknowledgements

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

CHAPTER 4

PSDNet Determination of Particle Size Distributions Using Synthetic Soil Images and Convolutional Neural Networks

Javad Manashti[1, 5], Pouyan Pirnia[1], Alireza Manashty[3], Sahar Ujan[4], Matthew Toews[2], François Duhaime[1]

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

2 Systems Engineering Department, École de technologie supérieure, Montreal, Canada

3 Computer Science Department, University of Regina, Regina, Saskatchewan

4 Electrical Engineering Department, École de technologie supérieure, Montreal, Canada

Paper submitted for publication, March 2022

4.1 Abstract

This article aimed to determine the grain size distribution of granular materials from images using convolutional neural networks. The application of ConvNet and pretrained ConvNet models, including AlexNet, SqueezeNet, GoogLeNet, InceptionV3, DenseNet201, MobileNetV2, ResNet18, ResNet50, ResNet101, Xception, InceptionResNetV2, ShuffleNet, and NASNetMobile was studied. Synthetic images of granular materials created with the discrete element code YADE were used. All the models were trained and verified with grayscale and color band datasets with image sizes ranging from 32 to 160 pixels. The proposed ConvNet model predicts the percentages of mass retained on the finest sieve, coarsest sieve, and all sieves with root-mean-square errors of 1.8 %, 3.3 %, and 2.8 %, respectively, and a coefficient of determination of 0.99. For pretrained networks, root-mean-square errors of 2.4 % and 2.8 % were obtained for the finest sieve with feature extraction and transfer learning models, respectively.

Keywords: Convolutional neural network; Pretrained network; Transfer learning; Particle size distribution; Discrete elements

4.2 Introduction

The particle size distribution (PSD) describes the statistical distribution of particle sizes in granular materials. Sieving (Gee and Bauder, 1986) is the standard method to determine the PSD of granular materials. The PSD characterizes the cumulative mass percentages passing a series of sieves with different opening sizes. Sieving is accurate, but it is time-consuming and difficult to adapt to online processing.

PSD determination is of great importance in most geotechnical engineering projects. For instance, materials used in construction projects such as embankment dams or roads are selected based on their PSD. Many soil properties, such as permeability, can be estimated based on the material PSD. The PSD also plays a primary role in the chemical and metallurgical industries (Ko and Shang, 2011a). Ore processing has PSD requirements (Costodes, Mausse, Molala, and Lewis, 2006; Schneider, Neumann, and Souza, 2007), and online PSD determination is fundamental for performance improvement (Ko and Shang, 2011b). It is also critical to detect and determine particle size distributions in the food (Ilonen et al., 2018), pharmaceutical (Caputo et al., 2019), and chemical industries (Allen, 2003).

Image-based methods have been used for PSD determinations for several decades. These methods provide a fast, non-contact, and economical way to determine the PSD, but they are not without disadvantages. Thurley and Ng (2008) divided the sources of error into segregation and grouping error, profile error, capturing error, and overlapping-particle error. These errors are caused by the photographs only capturing the surface of the material. Photographs are in effect “sampling” the material, and are unable to describe the whole sample.

Image analysis techniques can be divided into two groups: direct and indirect methods. Direct methods use image segmentation for PSD determination, while indirect methods extract textural features from the images which are then used to determine the PSD.

Most commercial codes use direct methods to determine the PSD. Various algorithms are used to detect the contours of each particle in the image. The PSD is estimated using the size of the segmented area and statistical principles. The accuracy of direct methods was appraised by Liu and Tran (Liu and Tran, 1996) by comparing the D_{50} obtained through sieving and a series of commercial software packages for mine backfill material. The D_{50} is the particle size for which 50 % of the mass if composed of smaller particles. Compared to sieving, the error on D_{50} was between 212 and 224 % for FragScan (Maerz, Palangio, Palangio, and Elsey, 2007), 32 and 60 % for WipFrag (Maerz et al., 1996), and between 37 and 48 % for Split (Girdner et al., 1996). The results presented by Liu and Tran (1996) can also be used to compute the root-mean-square error (RMSE) on the percentages passing for each sieve. For these three commercial codes, the RMSE varied between 13 and 36 %. Codes for particle size analysis in other fields, for instance BASEGRAIN in hydraulics and sediment transport, are also often based on segmentation (Detert and Weitbrecht, 2012).

The textural features used with indirect methods can be divided into three groups: statistical, local pattern, and transform-based features (Tian, 2013). Haralick features (Haralick et al., 1973), local entropy (Gonzalez and Woods, 2002; Khellaf, Beghdadi, and Dupoisot, 1991; , and histogram of oriented gradients (HOG) (Dalal and Triggs, 2005; Manashti, 2022) are based on statistics of pixel intensity. Features based on local patterns establish a relationship between the gray level of each pixel in an image and the pixels located in their neighborhood (Ojala et al., 2002). Local binary pattern (LBP) (Ojala et al., 2002), local configuration pattern (LCP) (Guo et al., 2011), and completed local binary pattern (CLBP) (Guo et al., 2010) are examples of local pattern features. Transform-based features describe the image texture in the frequency domain (Shin and Hryciw, 2004). These features include Haar wavelet transforms (Hryciw et al., 2015; Manashti, 2022; Yaghoobi et al., 2019), Fourier transforms (Szeliski, 2011; Yaghoobi et al., 2019), and Gabor filters (Tuceryan and Jain, 1993; Yaghoobi et al., 2019).

The textural features that are most commonly used for PSD determination in geotechnical engineering are Haar wavelet transforms. Hryciw et al. (2015) presented the theoretical basis for PSD determination with Haar wavelet transforms. These features form the basis of the Sedimaging system described by Ohm and Hryciw (2014a). This system combines mechanical grain size sorting through sedimentation in a water column with textural analysis of photographs of the material after sedimentation. Based on the PSD presented by Ohm and Hryciw (2014a) for clean sand, a RMSE on the percentages passing of 7.4 % can be calculated when comparing Sedimaging and sieving results.

The image features of indirect methods are often fed to artificial neural networks (ANN) to calculate the percentages passing or characteristic grain size (e.g. D_{50}). Ghalib et al. (1998) used Haralick features as inputs for an ANN to predict the particle size of uniform sands. Yaghoobi et al. (2019) used Fourier transforms, Gabor filters and wavelet transforms to generate inputs for a series of ANN to predict the PSD of fragmented rock material. Manashti (2022) compared the results obtained with 9 feature extraction methods covering statistical, local pattern and transform-based features to predict the PSD for synthetic images of granular materials. He obtained RMSE on the percentages passing varying between 4.8 to 6.6 % when using the features as inputs for a series of ANN. Some researchers also combined direct methods of PSD determination from photographs with ANN, for example by feeding statistics of segmented areas on a photograph to an ANN for PSD determination (Hamzeloo et al., 2014).

Deep learning methods, such as a convolutional neural networks (ConvNets) (Buscombe, 2020), can be considered as indirect methods for PSD determination. ConvNets (Abdel-Hamid et al., 2012; Chellapilla et al., 2006; Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; LeCun & Bengio, 1995) provide a general neural network structure for inputs consisting of images, sounds, and videos. Contrarily to the classical ANN methods described previously, the ConvNet processes the image itself via a layer-wise sequence of convolutions with trained filter banks separated by non-linear activation functions (e.g. rectification), where the convolution responses ultimately serve as the inputs of a fully connected neural network. A typical ConvNet consists of millions of weight parameters (i.e. filter coefficients) across a number of layers, hence the notion of a deep ConvNet. ConvNets serve as a general computational tool in virtually any image analysis context, including medical image analysis (Hall et al., 2010; Lee et al., 2017; Plautz et al., 2017; Tajbakhsh et al., 2016; Yan et al., 2018) and speech recognition (Abdel-Hamid et al., 2012, 2013; LeCun & Bengio, 1995). Well-known ConvNet architectures include AlexNet (Krizhevsky et al., 2012), SqueezeNet (Iandola et al., 2016) , GoogLeNet (Szegedy et al., 2015), InceptionV3 (Szegedy, Vanhoucke, et al., 2016), DenseNet201 (Huang et al., 2017), MobileNetV2 (Sandler et al., 2018), ResNet18 (Wu et al., 2018), ResNet50 (He et al., 2016) , ResNet101 (He et al., 2016), Xception (Chollet, 2016), InceptionResNetV2 (Ioffe and Szegedy, 2015; Szegedy, Ioffe, et al., 2016), ShuffleNet (Zhang et al., 2017), and NASNetMobile (Zoph et al., 2018). These architectures are typically trained on large numbers of generic image categories and objects, e.g. the ImageNet (Deng et al., 2009) dataset consisting of 1000 different object classes, allowing them to recognize wide variety of common objects, such as computers, cups, and animal species and to generalized to new objects.

There are only a few examples of ConvNet applications for particle size analyses (Buscombe, 2020; Lang et al., 2021; McFall et al., 2020; Pirnia et al., 2018). Pirnia et al. (2018) performed the first experiment using ConvNet for PSD determination in geotechnical engineering. Their ConvNet was also a preliminary version of PSDNet, the network that is introduced in this paper. A database containing more than 53 000 synthetic grayscale images of granular material was adopted. The images were obtained with the discrete element code YADE and had a resolution of 128×256 pixels The Microsoft Cognitive Toolkit (CNTK) (Yu et al., 2014) was used to train and test the model. The model included four convolutional layers followed by a rectified linear unit (ReLU) layer. Max pooling layers were used after the first two convolutional layers and after the final convolutional layer. Three dense layers were used after the ReLU layer. RMSE on the percentages passing of 6.9, 4.2, and 9.1 % were obtained for all sieves, only the finest sieve, and only the coarsest sieve, respectively.

Buscombe (2020) introduced SediNet, a ConvNet for the size classification of sediment images. PSD obtained from a manual segmentation of the images were used as the ground truth. SediNet consists of four convolutional blocks each having several two-dimensional convolutional filter layers, batch normalization layers, and two-dimensional max pooling layers. SediNet uses RGB images of 512×512 pixels. It was trained for three regression models. In the first regression model, SediNet used 205 images to predict 9 grain size percentiles and reach an RMSE ranging from 24 to 45 % of the mean size for each percentile. The second regression model was run on 31 images of beach sand with RMSE ranging from 16 to 29 % of the mean size for each percentile. The first and second methods predicted cumulative percentage passing in pixels. The third regression model predicted the sieve size directly in mm with an RMSE (normalized by the mean grain size) of 22 %.

Mcfall et al. (2020) conducted a study using four types of analyses to predict the grain size distribution of several beach sands collected by citizen scientists. They compared three image analysis techniques with sieve analyses: direct analyses based on image segmentation, transform-based features, and convolutional neural networks. The RGB images were gathered using cell-phone cameras. The final images had a size of 1024 × 1024 pixels. For the direct analysis, the grains in each image were identified by segmentation, and their size was estimated individually. Direct analysis by segmentation had a mean percentage error on D_{50} of 34.2%. This percentage error was reduced to 13.0% by drying the sample and by using a black background for the photograph. The Python implementation of the wavelet feature extraction method (pyDGS) proposed by Buscombe (Buscombe, 2020) was used to estimate the grain size distribution from textural features. The mean percentage error on D_{50} was 36.4%. For the ConvNet, they used the SediNet framework (Buscombe, 2020) to estimate the PSD. The mean percentage error on D_{50} was 22%.

GRAINet (Lang et al., 2021) is a ConvNet model to predict the grain size distribution of gravel bars from unmanned aerial vehicles (UAV) images. Lang et al. (2021) used 1,419 images from 25 gravel bars along six rivers in Switzerland with grain sizes ranging from 0.5 to 40 cm. As Buscombe (Buscombe, 2020), the ground truth consisted of PSD obtained from a manual segmentation of the images. Input images for GRAINet had a resolution of 500×200 pixels with normalized RGB channels. GRAINet includes a single 3×3 "entry" ConvNet layer, six convolutional blocks with batch normalization and RELU layers, and a 1×1 convolutional layer. Regression of the mean particle diameter (d_m) with this model leads to a RMSE of 27 % of the mean d_m value, where $d_m = \sum P_i d_i / 100$ with d_i and P_i as the mean size and percentage passing for bin I of the PSD.

While traditional neural networks can be trained properly with a few hundred samples, ConvNet usually require thousands of images to be trained (Goodfellow et al., 2016). The main dataset used to train SediNet included only 409 images. Lang et al. (2021) noted that SediNet appears to suffer from overfitting due to its small dataset. GRAINet was based on a larger dataset including 1,419 images. Different image augmentation methods can facilitate ConvNet training by increasing the number of images to avoid overfitting. Lang et al. (2021) flipped their images vertically and horizontally to increase their number. Mcfall et al. (2020) used 63 images and increased this number to 517 by flipping the images horizontally and by dividing each images into sub-images. Due to the resources needed to prepare large datasets for training ConvNet models, some researchers (Duhaime, Pirnia, Manashti, Temimi, and Toews, 2021; Frid-Adar et al., 2018; Rajpura, Bojinov, and Hegde, 2017) have tried to include synthetic images in their datasets. Duhaime et al. (2021) used three datasets comprising only real photographs, only synthetic images prepared with the discrete element method and a combination of both real photographs and synthetic images to train a series of ANN to predict the PSD from local entropy features. Similar features and performances were obtained with the three datasets. These results demonstrate that synthetic images can be used to generate rapidly a series of datasets simulating photographs with different characteristics (e.g. viewpoints, number of images, lighting conditions) to study their influence before creating the real dataset.

The main objective of this paper is to analyze the ability of ConvNet to predict the PSD of granular materials using synthetic images. The discrete element method was used to generate a large number of images corresponding to two viewpoints: from the top of a transparent container and from underneath. The performances of a ConvNet trained from scratch using our synthetic images was compared with the performances of generic pretrained ConvNets used as feature extractors, a process known as transfer learning. Color and grayscale images of different sizes were used to train and to evaluate the ConvNet. A large dataset of synthetic images was utilized to avoid overfitting and to show how synthetic images can be used a priori to verify the influence of different image parameters. This paper is the first to test pretrained ConvNet models for PSD determinations. Thirteen pretrained networks were compared in this study.

The dataset, the ConvNet structures, the pretrained networks, and transfer learning structure will be discussed in the methodology section. The results from each model and each dataset are compared in the result section.

4.3 Methodology

4.3.1 Preparation of synthetic granular image dataset

The discrete element code YADE was used by Pirnia et al. (2019) to create synthetic images of granular materials. A virtual box was filled with spherical particles with predetermined PSD and random colors. Color images were taken from the top (Figure 4.1a) and from under the transparent box (Figure 4.1b). For some image analysis methods, the color images were converted to grayscale (bottom row in Figure 4.1). Images were obtained for a wide range of PSD with particle sizes between 75 and 1180 µm. In terms of pixels, the particle sizes range from 12 to 106 pixels in the original 400 by 400 pixels images. The PSD were obtained by varying by 5 % increments the cumulative percentages passing for sieve sizes of 106, 150, 250, 425, and 710 µm. Each of the 53,003 top and bottom image pairs were used to obtain four views: Top (T, Figure 4.1a), Under (U, Figure 4.1Figure 2.18b), Top-Under (TU, Figure 4.1c), and Stretched Top-Under (STU, Figure 4.1d). Images in dataset TU combine the T and U images for a size of 800×400 pixels. The STU dataset contains the TU images resized to 400×400 pixels. Dataset STU was created to feed both views at once to the pretrained models as they require square images. The horizontally scaled images (STU) introduce artificial affine deformations to the particles and textures, but these deformations should not have an influence on the results as a specific network was trained for each dataset. The T, U and STU image datasets were downscaled to 32, 64, 96, 128, and 160 pixels for our model. Images of between 224 and 331 pixels were used to feed the pretrained networks.

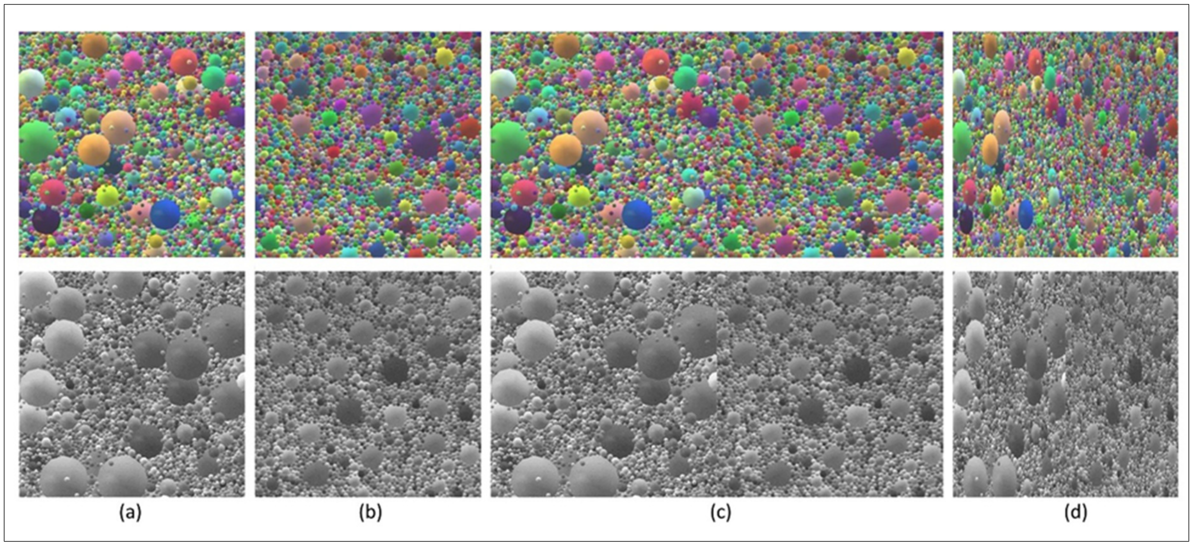

Figure 4.1 (a) Top, (b) Under, (c) Top-Under, and (d) Stretched-Top-Under views of the virtual granular material in a transparent box. Color images are shown in the first row. Grayscale images are shown in the second row

4.3.2 PSDNet

ConvNet structure is centered on convolutional layers. These layers extract feature maps from the images. The main difference between the textural features of indirect methods of PSD determination and ConvNet is that the later learns the weight and bias parameters of the filter kernels during network training. Non-linear functions such as ReLU and max pooling can be applied after the convolutional layers. Finally, fully connected layers can be followed by non-linear functions, including batch normalization, ReLU, and dropout, to predict the results.

Our proposed model is called PSDNet, a regression form of ConvNet. It predicts the percentage passing for a series of sieves directly from images. Different combinations of convolutional layers with various depths and settings have been assessed to obtain the best model of PSDNet to predict PSD. Images of 32, 64, 96, 128, and 160 pixels have been examined in both grayscale and RGB color modes to evaluate the effect of color and size of input image on the accuracy of PSDNet. MATLAB 2019b was used to define the architecture of the networks and for their training in this study.

4.3.2.1 Model name

To simplify the presentation of the methodology and results, abbreviations are used for the different dataset and model names. Each model name starts with G for grayscale images or C for color images. The number that follows gives the image size (32, 64, 96, 128, 160, 224, 227, 299, and 331 pixels). The name ends with the viewpoint and image type: T (Top), U (Under), TU (Top and Under), and STU (Stretched Top and Under). For example, dataset G128T includes grayscale images taken from the top of the virtual box with a size of 128×128 pixels. For model names ending with TU, the size in the abbreviation corresponds to the image height. The image width is twice the size given in the abbreviation. For example, model C32TU was trained with color images with views from the top and from under with a resolution of 32×64 pixels. For statements that are independent of image size, model names are used without image sizes, such as CU (color images from under) or GSTU (grayscale images with the stretched top and under views).

4.3.2.2 Structure of PSDNet

PSDNet consists of four convolutional blocks each with a set of convolutional layers followed by non-linear layers, including batch normalization, ReLU, max pooling, and dropout. The convolutional blocks are followed by three blocks comprising a fully connected layer followed by batch normalization, ReLU, and dropout layers. The structure of the convolutional and fully connected blocks will be described in the following sections. The architecture of PSDNet for image dataset G128TU is shown in Figure 4.2.

Convolutional layers have two main parameters: the number of filters and their size. The filter size represents the size of the kernel that sweeps the image (Lang et al., 2021). For example, a filter size of five pixels implies that a neighborhood of 5×5 pixels is used to calculate each value in the feature map. The number of filters determines the number of independent kernels that sweep the image, each with its own set of weights and its bias. Each filter produces its own layer, or feature map. For example, 64 output layers are created by applying 64 filters on an input layer. The number of filters controls the network depth. Lower filter numbers lead to shallower networks.

SediNet and GRAINet, the previous ConvNets for PSD determination, had different architectures. Lang et al. (2021) designed GRAINet based on one entry convolutional layer followed by six residual blocks (three convolutional blocks and three identity blocks). Each block contains three convolutional layers followed by batch normalization and ReLU layers. The network ends with one ConvNet layer, average pooling and SoftMax to classify the results. SediNet (Buscombe, 2020) consists of four convolutional blocks. Each block contains between 16 and 64 convolutional filters followed by batch normalization and max-pooling layers. The last convolutional block is followed by a dropout layer, a fully connected layer, and an output layer to predict the grain size percentiles. GRAINet and SediNet were respectively developed for 500×200 and 512×512 pixel color images.

PSDNet was designed based on four convolutional blocks that contain dropout layers in the second and fourth blocks. The final regression layer is preceded by three fully connected blocks before the regression output layer. Each fully connected block has a dropout ratio of 20 %. Previous models did not use dropout (GRAINet) or used it only in the fully connected block with a dropout ratio of 50 % (SediNet). Recently, dropout with a lower ratio (e.g. 20 % for PSDNet) have been used in the convolutional layer to avoid overfitting and increase model accuracy (Park and Kwak, 2017). The PSDNet structure was optimized for multi dimensional input images from 32 up to 160 pixels for both color and grayscale images.

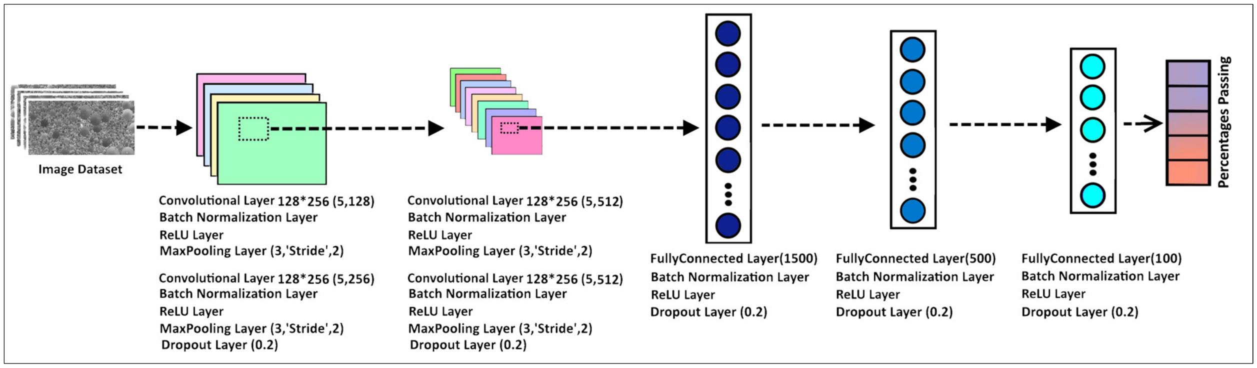
Image Dataset
Convolutional Layer 128*256 (5,128)
Batch Normalization Layer
ReLU Layer
MaxPooling Layer (3,'Stride',2)
Convolutional Layer 128*256 (5,256)
Batch Normalization Layer
ReLU Layer
MaxPooling Layer (3,'Stride',2)
Dropout Layer (0.2)
Convolutional Layer 128*256 (5,512)
Batch Normalization Layer
ReLU Layer
MaxPooling Layer (3,'Stride',2)
Convolutional Layer 128*256 (5,512)
Batch Normalization Layer
ReLU Layer
MaxPooling Layer (3,'Stride',2)
Dropout Layer (0.2)
FullyConnected Layer(1500)
Batch Normalization Layer
ReLU Layer
Dropout Layer (0.2)
FullyConnected Layer(500)
Batch Normalization Layer
ReLU Layer
Dropout Layer (0.2)
FullyConnected Layer(100)
Batch Normalization Layer
ReLU Layer
Dropout Layer (0.2)
Percentages Passing

Figure 4.2 PSDNet structure for G128TU. Four blocks of convolution layers are followed by batch normalization, ReLU, max poolling, and dropout layers. The convolutional blocks are followed by three fully connected layers and by batch normalization and ReLU layers

The influence of the number of filters on the performances of PSDNet was verified by training a series of models with different numbers of filters in the four convolutional blocks. Models G32STU, C32STU, and G32TU were chosen for the comparison. The number of filters in the first convolutional block was varied between 4 and 256. As shown in Figure 4.3, more than 32 filters were needed in the first block to reach the plateau corresponding to the lowest RMSE values on the percentages passing for models G32STU and C32STU. Similarly, a filter number equal to the input image size gave the best results for models with larger images (not shown here). Consequently, the filter number of the first layer was set equal to the input size. Larger numbers of filters could be used, but it increases the size of PSDNet and memory requirements for the Graphics Processing Unit (GPU), especially for larger images.

Preliminary tests with PSDNet have shown that better performances were obtained when the filter number was increased deeper in each network. Better performances were obtained when the number of filters was set as the image size for the first convolutional layer, double the image size for the second convolutional layer, and quadruple the image size for the third and fourth layers. For example, the number of filters for an image input size of 128 pixels was set at 128, 256, 512, and 512 for convolution blocks 1 through 4 (Figure 4.2). Buscombe (Buscombe, 2020) also used four blocks with increasing filter numbers for SediNet (16, 32, 48, and 64 for the first to last blocks).

Figure 4.4 shows the effect of filter size on the performances of three models. The filter sizes with the minimum RMSE on the percentages passing are 3, 4, and 5. Filters of 5×5 pixels were selected for the PSDNet convolutional layers. A padding of size 2 was applied around the inputs (image and feature maps) to maintain a constant output size. The inputs were padded with zeros.

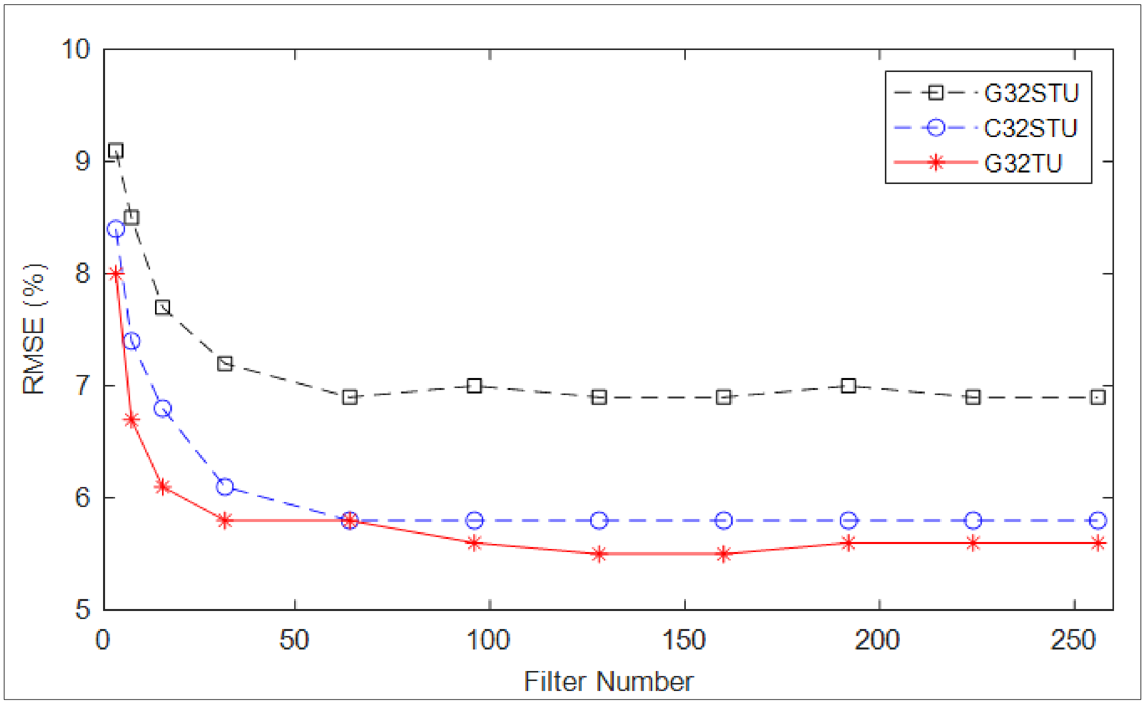

Figure 4.3 Effect of filter number on the percentage passing RMSE when filter size is set to five

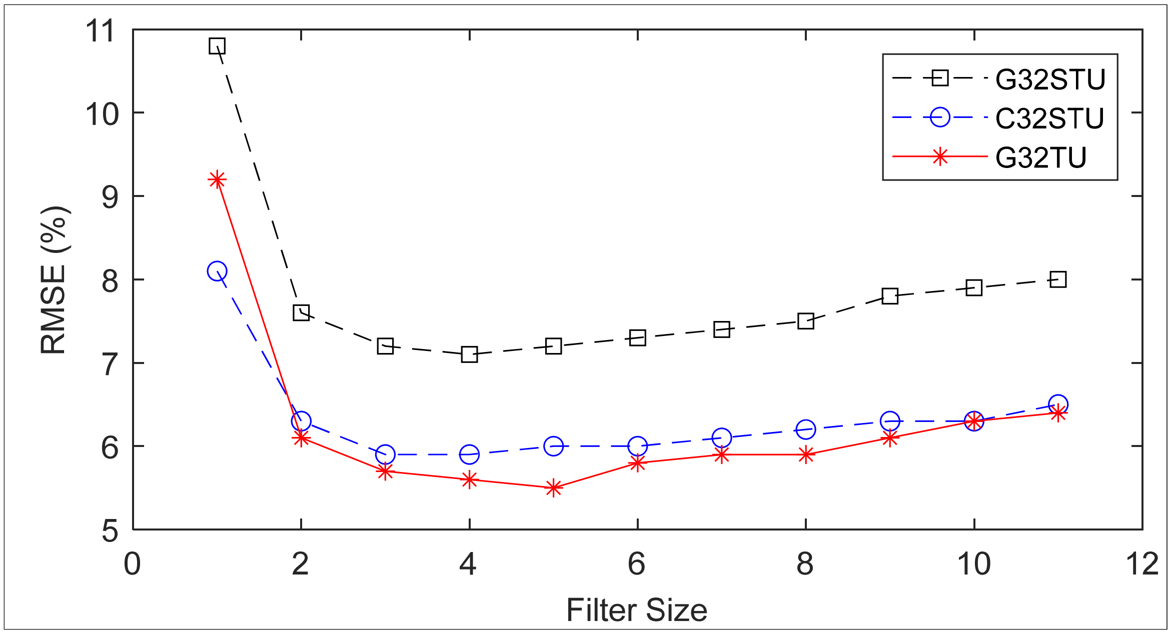

Figure 4.4 Effect of filter size on percentage passing RMSE when filter number is set to 96, 32, and 32 for models G32TU, G32STU, and C32STU, respectively

Batch normalization layers set the mean and variance of the output of the previous layer (Goodfellow et al., 2016). This function can be used to speed up training and reduce the sensitivity to network initialization data (Ioffe and Szegedy, 2015). Typically, batch normalization is used after each convolutional layer and before other non-linear functions such as the Rectified Linear Unit (ReLU) layer.

ReLU layers are used regularly after batch normalization layers (Dahl, Sainath, and Hinton, 2013). These layers intensify the model non-linearity by replacing all the negative activations with zero (Goodfellow et al., 2016). The activation function of ReLU layer is defined as:

$f(x) = \max(0, x)$ (1)

where f(x) is the threshold function, and x is the function input.

Max pooling layers (Scherer, Müller, and Behnke, 2010) are added to the network to reduce the data size, computation, and overfitting (Buscombe, 2020). Max pooling separates the input layer into smaller pooling areas and finds the maximum activation for each area. Max pooling is applied to n×n pooling areas. In PSDNet, the max pooling layer is applied with a 3×3 filter size and a 2×2 stride. Stride is the movement of the pooling region in both vertical and horizontal directions. A stride of more than 1 reduces the size of the layer output with respect to the input. In PSDNet, the pooling regions overlap because the stride is less than the pooling dimension.

Dropout layers are used after every two packages of convolutional block, including convolutional layer, batch normalization, ReLU, and max pooling. To avoid overfitting of training data, dropout layers can be used to change some of the activations to zero randomly (Park and Kwak, 2017; Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 2014). In PSDNet, dropout layers were used so that 20 % of the number of input layers was set to zero.

Next, three fully connected (dense) layers followed by ReLU and dropout layers are applied. Fully connected layers function like normal neural networks and connect all activations from the previous layer (Karpathy, n.d.). The last layer in PSDNet is linear. This layer provides a regression output to predict the percentages passing for each sieve.

The total number of learnable parameters for PSDNet varied from 1.6 million for G32T up to 182 million for C160TU. As a comparison, GRAINet (Lang et al., 2021), ShuffleNet (Zhang et al., 2017), ResNet101 (He et al., 2016) and vgg19 (Simonyan and Zisserman, 2014) including 1.6 million, 14 million, 44 million, and 144 million parameters, respectively.

4.3.2.3 Network training

For training PSDNet, Stochastic Gradient Descent with Momentum (SGfDM) was used as a solver. The best PSDNet results were obtained when the initial learning rate was set at 0.0001. Different minibatch sizes were used based on the image size and GPU memory. For example, the minibatch size was set at 256 for G32T and 10 for C160TU. The minibatch size was limited by GPU memory (NVIDIA GeForce RTX 2080 with 8 GB memory and NVIDIA GeForce GTX 1650 with 4 GB memory). The minibatch size corresponds to the number of images that are used to calculate the gradient of the loss function and to update the weight. The training and validation datasets were shuffled before each training epoch. Due to the large training (42 403 images) and validation (5300 images) datasets, the training and validation process took about 1 hour for grayscale images with a size of 32 pixels and up to 100 hours for the color images of 160 pixels. The complete dataset was divided in training, validation and testing subdatasets according to proportions of 80/10/10 %, respectively (Chakraborty et al., 2019; Gholamy et al., 2018).. Percentages passing were predicted for five sieves (106, 150, 250, 425, and 710 µm).

Figure 4.5 shows the effect of the number of training epochs on the RMSE of the test dataset. For C128T, the best RMSE on the percentages passing is obtained when the number of epochs is equal to 7.

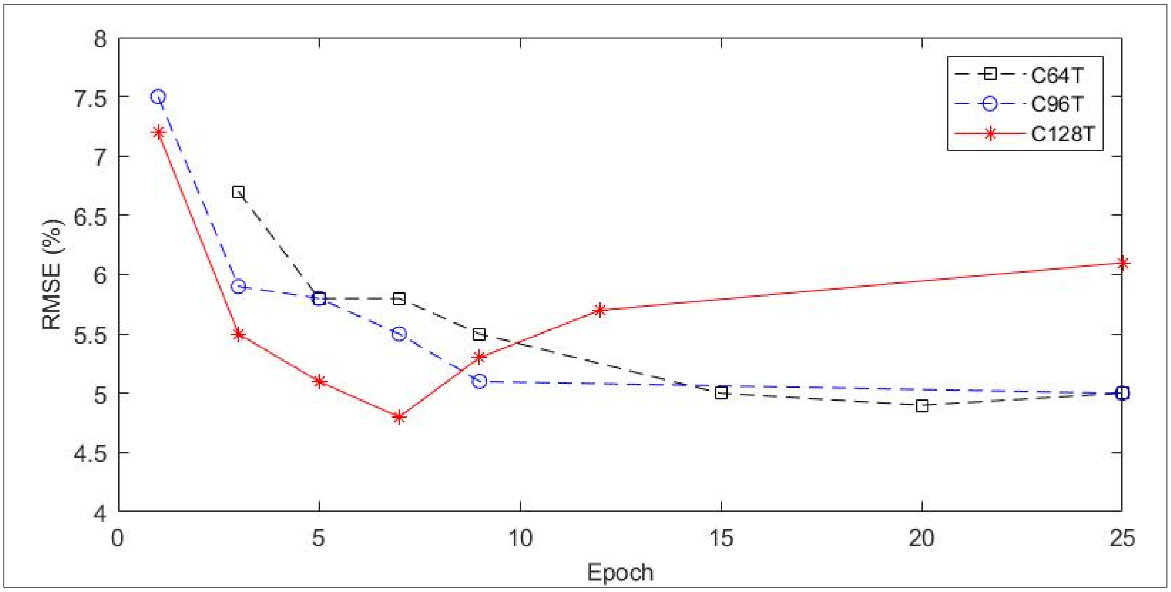

Figure 4.5 Effect of number of epochs on the RMSE of the percentage passing for all sieves and the test dataset

4.3.3 Pretrained ConvNets

Organizing thousands of labeled images is not possible for all applications as such datasets can be rare, expensive, or inaccessible (Weiss, Khoshgoftaar, and Wang, 2016). In this case, high-performance models can be pretrained with an accessible dataset for a different application and used for transfer learning. For example, networks trained on the ImageNet (Deng et al., 2009) dataset for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015) can be partially retrained or used as feature extractors for other applications and datasets.

4.3.3.1 Feature extraction

The feature extraction method is a quick way to use pretrained ConvNet. It does not require extensive computational power. This method is more useful when the dataset size is small. In the feature extraction method, the pretrained networks are used to extract features from the last layer. These features are fed to a new ANN to predict the percentages passing. With this method, the original pretrained network is not trained with the granular material images.

Twelve networks were compared to evaluate the performances of pretrained networks as feature extractors to estimate the PSD of granular materials. The networks include AlexNet, SqueezeNet, GoogLeNet, InceptionV3, DenseNet201, MobileNetV2, ResNet18, ResNet50, ResNet101, InceptionResNetV2, ShuffleNet, and NASNetMobile. The last layer of each pretrained network was used to extract 1000 features for the T, U, and STU images, separately. For the TU images, 2000 features were extracted by combining the features from the T and U images. The size of the input images for these networks ranges between 224 and 331 pixels depending on their structure. It should be noted that the input size of the pretrained networks is larger than the input size used with PSDNet (32 to 160 pixels).

The PSD was predicted with a neural network with one hidden layer with 10 neurons. The ANN was trained with MATLAB. For the training function, the Levenberg-Marquardt algorithm was set. The complete dataset was divided between the training, validation and testing subdatasets with proportions of 70/15/15 %, respectively (Chakraborty et al., 2019; Gholamy et al., 2018)..

4.3.3.2 Transfer learning

With transfer learning, the last layer of the pretrained networks can be replaced by a new regression layer and the models can be trained again with the new dataset (Weiss et al., 2016). This method is more computationally intensive than feature extraction as the complete pretrained networks are trained starting from the weight obtained during training with the primary dataset (e.g. ImageNet). Nevertheless, transfer learning is a faster than designing and training a new ConvNet model for large datasets. In this study, 12 pretrained networks including AlexNet, GoogLeNet, InceptionV3, DenseNet201, MobileNetV2, ResNet18, ResNet50, ResNet101, Xception, InceptionResNetV2, ShuffleNet, and NASNetMobile were trained to predict the PSD.

The classifier layer (softmax) was replaced with a regression layer to predict the PSD. The networks were trained for 5 epochs. Images from the T, U and STU datasets were used to train the networks. Since the ConvNet were pretrained for square images, it was not possible to evaluate the TU dataset with this method.

4.4 Results and discussion

4.4.1 Influence of image size and color

Images of 32, 64, 96, 128, and 160 pixels were examined in both grayscale (G) (Figure 4.6, top row) and RGB (C) (Figure 4.6, bottom row) modes to understand the effect of image input size and color on the ConvNet results. Four viewpoints and image stitching methods were compared: top (T), under (U), top and under (TU), and stretched top and under (STU).

As shown in Figure 4.6, the RMSE on the percentages passing for all sieves generally decreases when the image size increases from 32 to 160 pixels in both grayscale and color modes. The image combining the top and under viewpoints has the best results for both color and grayscale images, followed by stretched top and under, top, and under views, respectively. For example, the RMSE for the GTU datasets goes from 5.8 to 3.1 % for 32 and 160 pixels images, while for GU it goes from 8.4 to 4.8 % for the same sizes. It is worth noting that the size of TU images is doubled compared to other methods. STU images achieve better results compared to the top or under views with the same size, even if the images are stretched.

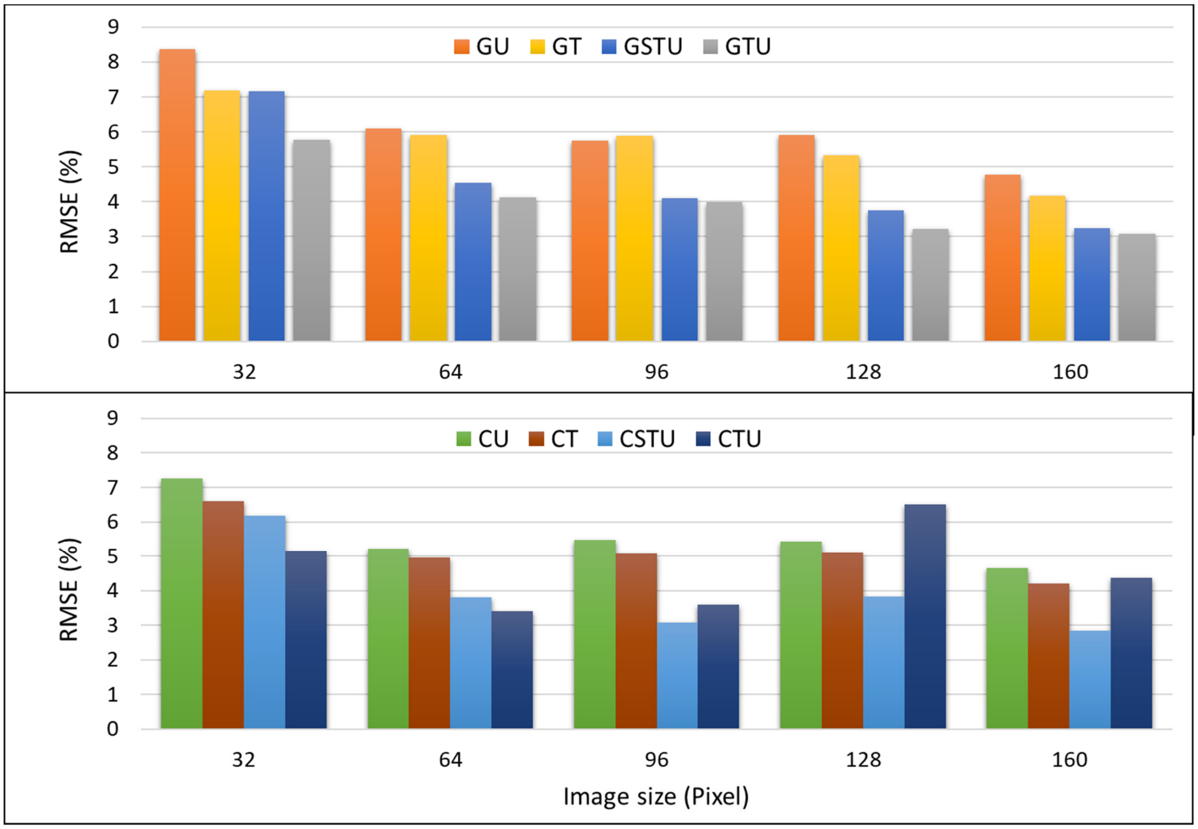

Figure 4.6 Effect of PSDNet image size on RMSE in grayscale (top row) and color (bottom row) images for different capture views of Under (U), Top (T), Stretched Top-Under (STU), and Top-Under (TU) datasets

Most of the time, the RMSE is lower when using color images compared to grayscale images. The RMSE reaches 2.8% for C160STU. Color datasets can reach better results than grayscale datasets with smaller images. For example, model C96STU has a similar RMSE to model G160TU. As for the grayscale images, TU models have better results followed by STU, T, and U for RGB views, respectively. The RMSE difference between grayscale and color modes is reduced by increasing the image size. For example, C32U is better than G32U by more than 1%, but C160U is only 0.1% better than G160U.

The previous ConvNet model trained by Pirnia et al. (Pirnia et al., 2018) achieved RMSE on the percentages passing of 6.9, 4.2, and 9.1 % for all sieves, the finest sieve, and the coarsest sieve, respectively. As mentioned previously, it was trained using grayscale images of 128×256 pixels. With the same dataset, PSDNet could reach RMSE values of 3.2, 1.8, and 3.0 % for all sieves, the finest sieve, and the coarsest sieve, respectively. The version of PSDNet trained with datasets C128STU and C160STU had an RMSE value of 1.6% for fine particles and 2.8 % for all sieves, respectively. This highlights the importance of optimizing the network architecture and parameters.

The influence of using a synthetic dataset for ConvNet training instead of real photographs is difficult to appraise. Duhaime et al. (2021) compared the RMSE on the percentage passing for a series of ANN trained using local entropy features and different datasets with the same PSD in terms of pixels per diameter. They obtained a RMSE of 3.4 % for synthetic images similar to the grayscale images used in this paper, 3.8 % for real photographs and 4.4 % when combining both datasets. Therefore, even if better performances can be expected from synthetic dataset, the difference with real photographs is relatively small. It should be noted that synthetic datasets have been used for ConvNet training in other fields (e.g. Nikolenko, 2021; Rajpura et al., 2017).

Manashti (2022) used the same synthetic images to compare nine indirect methods based on traditional features used as inputs for a series of ANN. The best method achieved RMSE values of 3.4, 1.7, and 4.7 % for all sieves, the finest sieve, and the coarsest sieve, respectively. In comparison, PSDNet produced RMSE values of 3.2, 1.8, and 3.0 % for the same dataset (G128TU). Thus, methods based on traditional feature extraction obtained similar results for finer particles, but better results were obtained with ConvNet for larger particles. Most methods based on traditional feature extraction performed relatively poorly for coarser particles. This could be a significant advantage of ConvNet.

PSDNet achieved good results compared to those reported in the literature for real datasets and direct methods based on image segmentation. RMSE of respectively 13-36 % and 15-20 % can be calculated from the results presented by Liu and Tran (1996) and Sudhakar et al. (2006), respectively. Both studies were based on the commercial software package WipFrag. It should be noted that training and running ConvNet requires much more computational power and much larger datasets than the conventional direct and indirect methods. Training ConvNet with a thousand images or less is likely to lead to overfitting problems (Lang et al., 2021). With smaller number of images, the direct method used in commercial software, indirect methods based on feature extractions, and pretrained ConvNet are more suitable. Combining synthetic and real datasets as done by Duhaime et al. (2021) might be an efficient solution to train ConvNet networks with only a few thousands real photographs.

4.4.2 Pretrained feature extraction

Figure 4.7 presents the results for pretrained ConvNet used as feature extractors with the TU dataset. InceptionResNetV2 achieved the best results with an RMSE value of 3.6 % for all sieves. In addition, a combination of the features extracted from the MobileNetV2, ResNet101, DenseNet201, and InceptionResNetV2 networks (8000 features) reached a RMSE value of less than 3 % for fine particles.

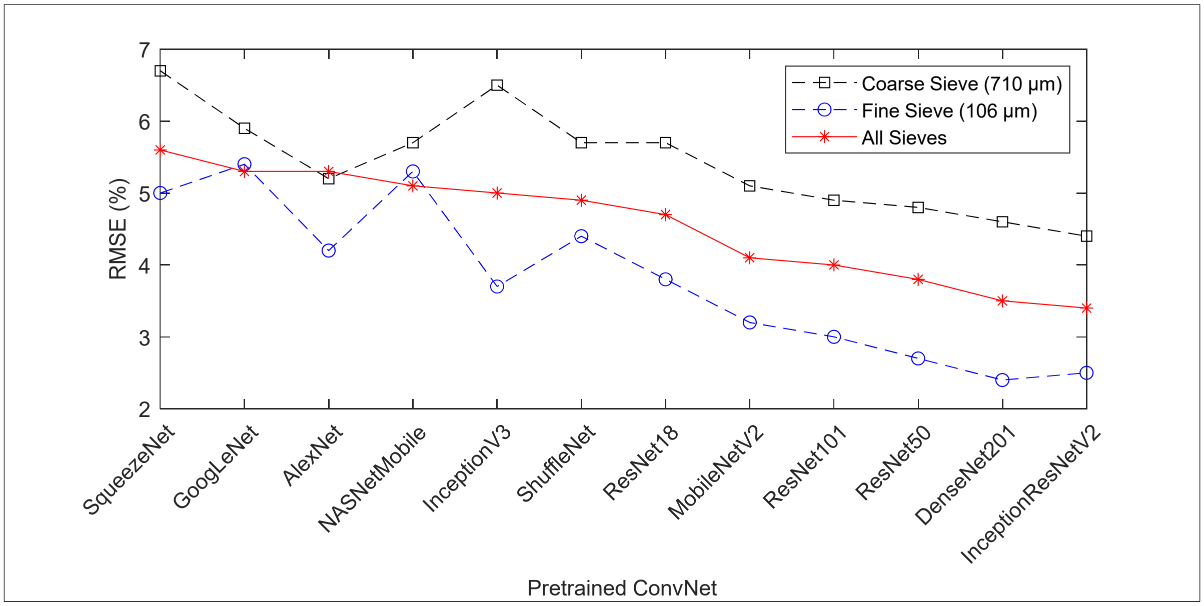

Figure 4.7 Performances of pretrained ConvNet models used as feature extractors on the prediction of all sieves, coarse, fine sieves for TU view

Figure 4.8 represents the performances of different pretrained models used as features extractors to predict the PSD for the T, U, TU, and STU datasets. Like PSDNet, pretrained ConvNet also achieved better results for TU images followed by STU, T, and U. InceptionResNetV2 achieved an RMSE value of 3.4 % for TU images as the best result. The highest RMSE was obtained with SqueezeNet (7.5 %) for the U dataset. The pretrained models used the color images of 224 to 331 pixels. Compared to PSDNet, the best pretrained feature extractor achieved results similar to network C64TU. Network G128TU achieved better results than InceptionResNetV2 for TU images.

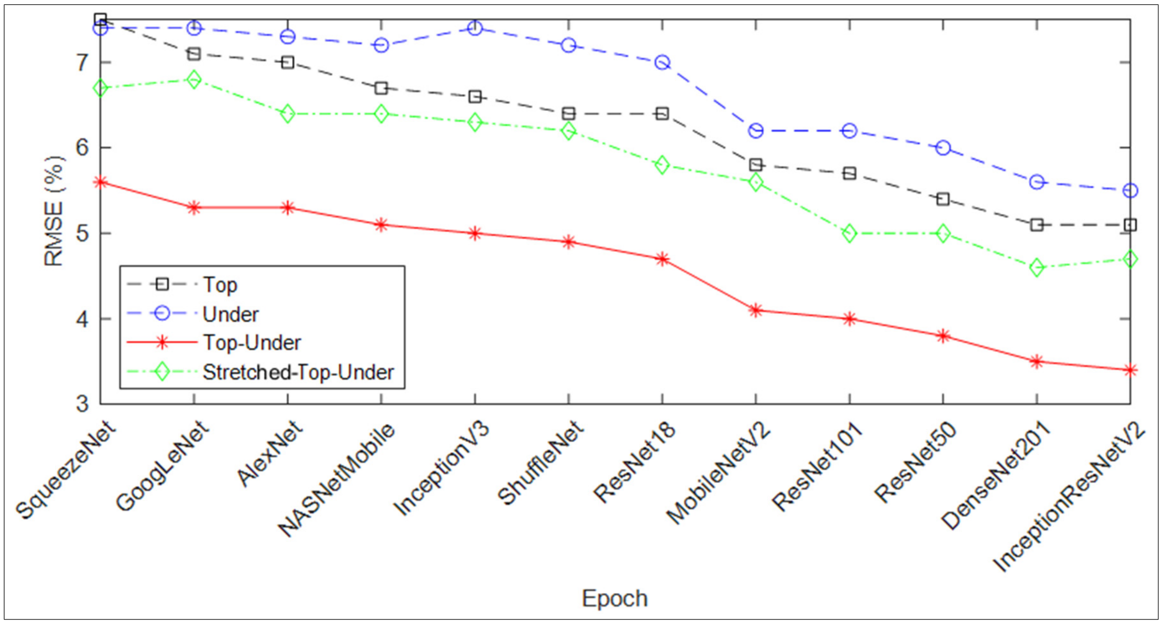

Figure 4.8 Effect of image viewpoint and stitching on PSD prediction with 12 pretrained models used as feature extractors

The results for pretrained feature extractors are similar to the results reported by Manashti (2022) for nine different traditional feature extractors. For fine particles, DenseNet201 predicts the percentages passing with an RMSE of 2.4 % while LCP got an RMSE of 1.9% as the best traditional feature extractor. The pretrained models predict the PSD for the TU dataset with an RMSE value ranging from 3.4 to 5.6 %, while the RMSE values for traditional feature extraction ranges from 3.4 % to 6.9 % when all sieves are considered. It should be noted that the pretrained models reach these results using 1000 or 2000 features whereas the traditional models use between 7 and 3880 features. In comparison, Liu and Tran (1996), achieved RMSE values ranging from 13 to 36% for backfill muck pile with FragScan, WipFrag, and Split. Sudhakar et al. (2006), in other projects, reached an RMSE value of 15 % for WipFrag and 20 % for Fragalyst for the blasted sandstones. Both papers by Liu and Tran (1996) and Sudhakar et al. (2006)used real photographs.

4.4.3 Transfer learning

With the transfer learning method, the last classification layer of the pretrained ConvNet models was replaced with a regression layer. The model parameters were fine-tuned starting from the weight and bias values learnt during pretraining with the ImageNet database. Figure 4.9 shows the RMSE on the percentages passing for all sieves after training for five epochs with the T, U and STU datasets. Transfer learning models achieved their best results with the STU views with RMSE values ranging from 3.6 to 5.7 %. In contrast with the feature extraction method presented in the previous section, similar results were obtained with the T and U datasets, with the U dataset performing better for half of the pretrained models.

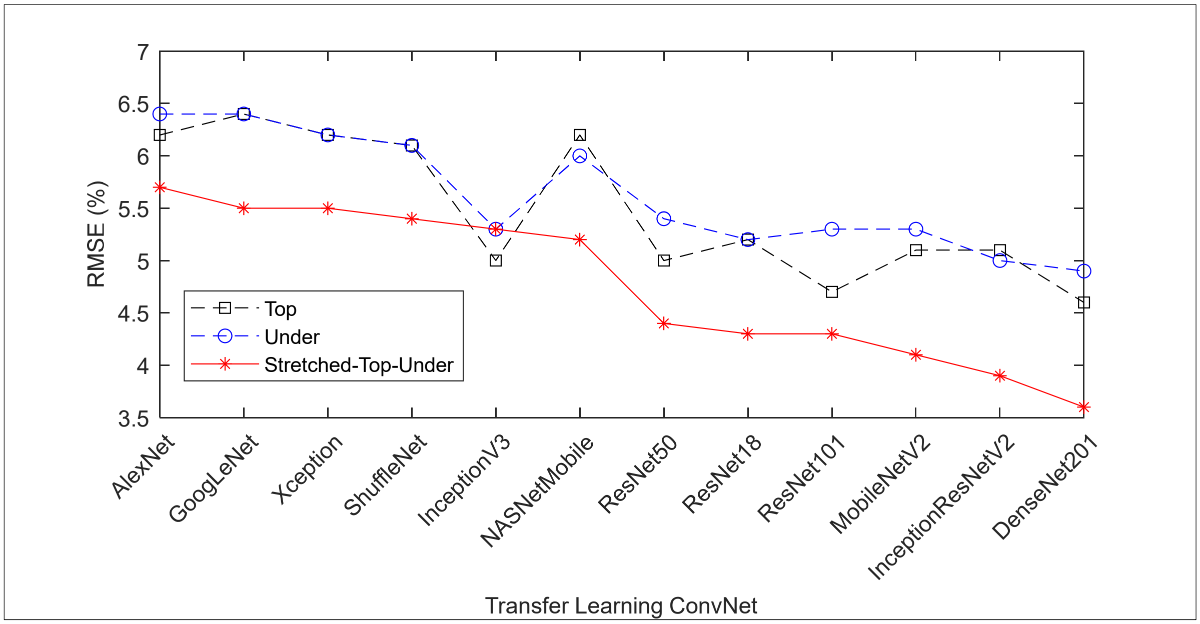

Figure 4.9 Results of transfer learning of 14 pretrained ConvNet models for T, U, and STU datasets

Compared to the feature extraction method, the transfer learning method gives better results when comparing the results for the same dataset. On the other hand, transfer learning cannot run on the TU dataset, while feature extraction achieves good results on the same dataset. InceptionReseNetV2 as a feature extraction method achieved an RMSE of 3.4 % for the Top-Under view, while DenseNet201, the best method of transfer learning, achieved an RMSE of 3.6 % for the STU view.

The results presented by Manashti (2022) for a selection of 618 traditional features are slightly better than the results obtained with transfer learning in this paper. Some of the transfer learning methods presented in this paper have better results than the individual feature types compared by Manashti (2022) (HOG, Wavelet, or Fourier) with RMSE values ranging from 6.2 to 6.9 % for all sieves.

4.4.4 PSD chart

Figure 4.10 compares the real and predicted percentages passing for a sample of six randomly selected synthetic soil images from the C160STU dataset. The percentages passing are plotted on the usual semi-log plot. The blue diamonds and lines indicate real PSD, while the red circles and lined indicate the predictions. Overall, the real and predicted grain size distributions are similar. It should be noted that PSDNet sometimes predicts PSD that are physically impossible. For example, in Figure 4.10e, the percentage passing decreases from 425 to 710 µm.

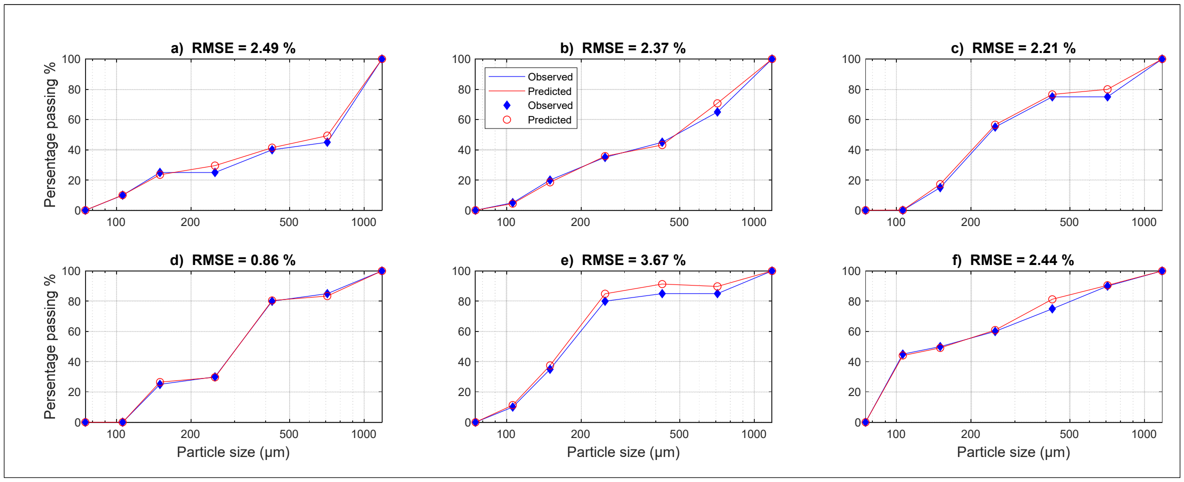

Figure 4.10 Six randomly selected samples of real and predicted PSD obtained with PSDNet for dataset C160STU

4.4.5 D_{50}

The D_{50} can be estimated using a linear interpolation between the two PSD data points below and above a percentage passing of 50%. Figure 4.11 represents the prediction and observation of the percentage error of D_{50} for the C96STU dataset. The mean error percentages on D_{50} for model C96STU was 5.6%. Slightly higher error percentages were obtained for larger images with transfer learning (6.0%) and pretrained feature extraction (6.0%). Quite similar results on error percentages were achieved for traditional feature extraction (6.1%)(Manashti, 2022). Mcfall et al. (2020) obtained an error of 22 % on the D_{50} of beach sand with SediNet, the ConvNet presented by Buscombe (2020). For rock fragmentation, the mean D_{50} error was 55 % and 100 % for WipFrag and Split, and FragScan, respectively (Q. Liu and Tran, 1996).

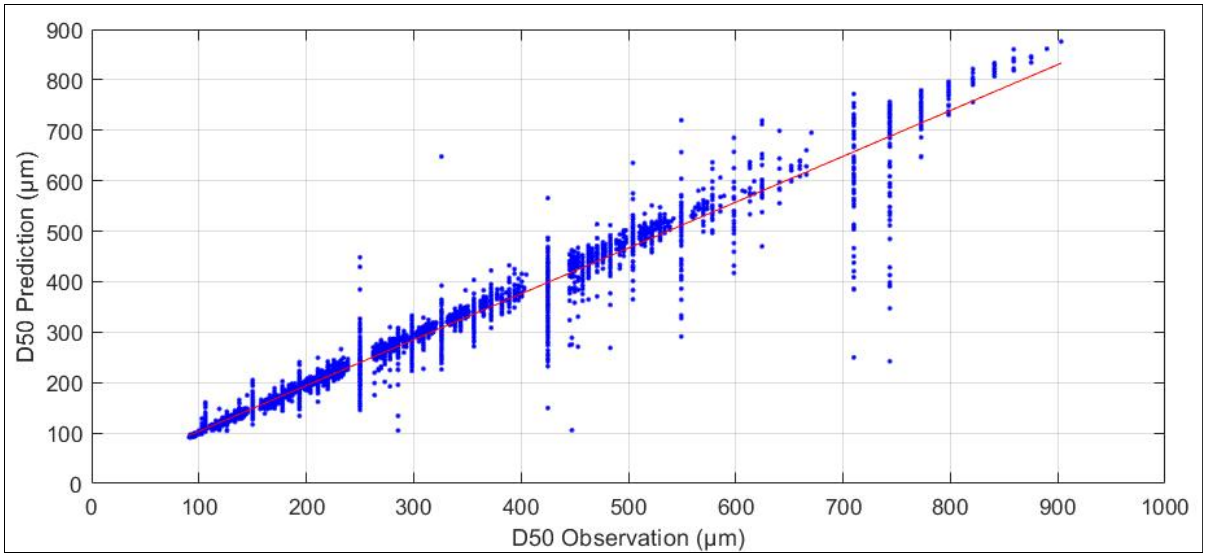

Figure 4.11 Predicted and observed D_{50} for model C96STU

4.5 Conclusion

The primary aim of this study was to use ConvNet to predict the PSD of granular material using a large dataset of synthetic images. The results show the ability of ConvNet to predict the PSD from images even though some particles are fully or partly covered by others. This new method can provide fast, accurate, and economical online PSD predictions in the laboratory or in the field.

The best performances were obtained with PSDNet (model C160STU) with RSME values of 1.8 % for fine particles, 3.3 % for coarse particles, and 2.8 % for all sieves. Manashti (2021) presented results for the same dataset with traditional feature extraction methods. His combined feature extraction method reached RMSE values of 1.7, 4.7, and 3.4 % for fine particles, coarse particles and all sieves, respectively. The performance improvement with ConvNet for coarser sieves is significant as traditional feature extraction tended to perform poorly for coarse particles.

This paper was the first to use pretrained ConvNet for feature extraction or transfer learning to determine the PSD of granular material. Several pretrained models were examined in this paper using our synthetic image dataset. For feature extraction of pretrained ConvNet, an RMSE value of 3.4 % was obtained for all sieves and the STU dataset. This RMSE is identical to the value obtained by Manashti (2021) for the same dataset and a combination of traditional feature extraction methods. Transfer learning did not improve the results compared to pretrained networks used as feature extractors or traditional feature extraction (Manashti, 2022).

The role played by training is one of the main differences between direct methods of PSD determination based on segmentation and indirect methods based on textures and neural networks. Training allows the network to take into account the hidden particles implicitly. Methods based on segmentation only analyze the particles at the surface and must account for hidden particles explicitly through statistical relationships. Because they are machine learning models, ConvNet can also be improved through operation by expanding the dataset.

4.6 Acknowledgments

Training of the preliminary version of PSDNet introduced by Pirnia et al. (Pirnia et al., 2018) was made possible by a Microsoft Azure sponsorship. This project was supported by Hydro-Québec and the National Sciences and Engineering Research Council of Canada (NSERC).

CHAPTER 5
Comparing PSDNet, pretrained networks, and traditional feature extraction for predicting the particle size distribution of granular materials from photographs

Javad Manashti[1], François Duhaime[2], Matthew F. Toews [3], Pouyan Pirnia[4], Jn Kinsonn Telcy[5]

Paper submitted for publication, March 2022

5.1 Abstract

This study aims to evaluate PSDNet, a series of convolutional neural networks (ConvNets) trained with photographs to predict the particle size distribution of granular materials. Nine traditional feature extraction methods and 15 pretrained ConvNets were also evaluated and compared. A dataset including 9600 photographs of 15 different granular materials was used. The influence of image size and color band was verified by using six image sizes between 32 and 160 pixels, and both grayscale and color images as PSDNet inputs. In addition to random training, validation, and testing datasets, a material removal method was also used to evaluate the performances of each image analysis method. With this method, each material was successively removed from the training and validation datasets and used as the testing dataset. Results show that a combination of all PSDNet color and grayscale features can lead to a root mean square error (RMSE) on the percentages passing as low as 1.8 % with a random testing dataset and 9.1% with the material removal method. For the random datasets, a combination of all traditional features, and the features extracted from InceptionResNetV2 led to RMSE on the percentages passing of 2.3 and 1.7 %, respectively.

[1] Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, Montréal, Quebec, H3C 1K3, Canada; https://orcid.org/0000-0001-9879-3931

[2] Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, Montréal, Quebec, H3C 1K3, Canada; https://orcid.org/0000-0001-8289-5837

[3] Systems Engineering Department, École de technologie supérieure, Montréal, Quebec, H3C 1K3, Canada; https://orcid.org/0000-0002-7567-4283

[4] Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, Montréal, Quebec, H3C 1K3, Canada; https://orcid.org/0000-0003-2939-2813

[5] Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, Montréal, Quebec, H3C 1K3, Canada

Keywords: PSDNet, Convolutional Neural Network, Feature extraction method, Pretrained ConvNet, Particle size distribution

5.2 Introduction

In geotechnical engineering, the particle size distribution (PSD) of particles larger than 0.075 mm is usually determined through sieving, according to ASTM D6913, *Standard Test Methods for Particle-Size Distribution (Gradation) of Soils Using Sieve Analysis.* Sieving is one of the most common tests conducted in commercial laboratories. However, sieving is time-consuming and energy-intensive. It is also comparatively expensive with regards to equipment maintenance and replacement of defective or worn sieves. The volume of water required for washing is also high (Ohm 2013). Conversely, image-based methods are eco-friendly, sustainable, and quick, with no need for water and low energy consumption. The benefits of PSD determination by image analysis over sieving include improved laboratory environment, reduced energy consumption, and shorter test time (Ohm 2013).

Image analysis methods for PSD determination can be divided in two main categories: direct and indirect methods. Direct methods use image segmentation. The contour of each particle in the photograph is traced. Statistical relationships are used to relate the size distribution of the segmented areas with the PSD. Commercial codes such as WipFrag (Maerz, Palangio, and Franklin 1996) are often centered on direct methods. Indirect methods use features that describe image texture to estimate the PSD. Textural features describe the spatial structure of pixel intensity (Liu et al. 2019; Tuceryan and Jain 1993).

Textural features can be classified into three main categories (Tian, 2013). The first category uses statistical parameters to describe the spatial distribution of pixel intensity. For example, Haralick features (Haralick, Shanmugam, and Dinstein 1973) use statistics (e.g., average, variance, correlation) of a global co-occurrence matrix that defines the number of pixel pairs separated by a specific offset for each combination of pixel intensity. Features of the second category are based on local patterns found in the image. One example is the Local Binary

Pattern (LBP) method (Ojala, Pietikainen, and Maenpaa 2002). With this approach, the relative intensity of pixels in the neighborhood of a center pixel are encoded as a binary integer. Transform-based methods form the third category. Gabor wavelets (Manjunathi & Ma, 1996) and Fourier transforms (Szeliski, 2011; Yaghoobi et al., 2019) are included in this category. Transformation-based methods allow the extraction of information on the distribution of pixel intensity in the frequency domain (Shin & Hryciw, 2004).

Several studies have appraised the performances of PSD determination methods based on textural features. Using a dataset of 53,003 synthetic images of granular material prepared with the discrete element code YADE (Pirnia et al. 2019), Manashti et al. (2021) reviewed the performances of nine textural parameters from the three main categories. The textural features were used as inputs for a series of artificial neural networks. All nine methods could perform relatively well with idealized spherical particles, pending an optimization of their parameters. A selection of the 618 best features from the three main categories gave a root mean square error (RMSE) on the percentages passing of 3.4 % and a relative error on the D_{50} of 6.1 %, where D_{50} is the particle size for which 50 % of the sample mass is composed of smaller particles. Performances with real photographs can also be estimated from published data. Ohm and Hryciw (2014a) used a semi-empirical relationship based on Harr wavelet transforms to estimate the PSD of granular material from photographs after mechanical sorting in a sedimentation column. An RMSE of 7.4 % for the percentages passing and a relative error of 7.6 % for the D_{50} can be estimated from their results.

Convolutional neural networks (ConvNets) provide a new artificial neural network structure for inputs consisting of images and sounds (Abdel-Hamid et al. 2012; Chellapilla, Puri, and Simard 2006; Kavukcuoglu et al. 2010; Krizhevsky, Sutskever, and Hinton 2012; LeCun and Bengio 1995). ConvNets apply a series of filters on the images to extract deep data that is fed to a fully connected neural network. A typical ConvNet uses a large number of layers with millions of weight parameters and thousands of training images and tuning parameters (Lee et al., 2017). Contrarily to the method used by Manashti et al. (2021) with traditional textural features fed to a fully connected network, ConvNets do not use engineered features but automatically determines the best features from the images to minimize the error rate (Fu & Aldrich, 2018). ConvNet are used in many areas, including medical imaging (Hall et al. 2010; Lee et al. 2017; Plautz et al. 2017; Tajbakhsh et al. 2016; Yan, Lu, and Summers 2018), speech recognition (Abdel-Hamid, Deng, and Yu 2013; Abdel-Hamid et al. 2012; LeCun and Bengio 1995), and soil classification and geochemistry (Liu, Ji, and Buchroithner 2018; Ng, Minasny, and McBratney 2020; Padarian, Minasny, and McBratney 2019). The most well-known ConvNet models were developed with the ImageNet dataset (Deng et al. 2009) to identify common objects such as computers, cups, and animals. The performances of ConvNets are generally superior to conventional approaches based on engineered features (Fu & Aldrich, 2018).

Pirnia, Duhaime, and Manashti (2018) made the first attempt at using ConvNets to predict the PSD of granular materials based on images in geotechnical engineering. They used the same synthetic image dataset as Manashti et al. (2022). Grayscale images of 128×256 pixels were used. The model structure was composed of four ConvNet blocks with 5×5 filters followed by a rectified linear unit (ReLU) layer. Each pair of convolutional blocks was followed by a max pooling layer. The four convolutional blocks were followed by three fully connected layers. The RMSE for all sieves, the finest, and coarsest sieve were 6.9, 4.2, and 9.1 %, respectively. The ConvNet model presented by Pirnia, Duhaime, and Manashti (2018) was later improved and forms the basis of PSDNet, the model presented in this paper. Compared to Pirnia, Duhaime, and Manashti (2018), PSDNet was improved by increasing the number of neurons in the fully connected layers, by using max pooling layers after each convolutional block, by adding dropout layers in the convolutional blocks and by increasing the number of filters in the convolutional blocks. The influence of image size (32 to 160 pixels), color band (color or greyscale image) and viewpoints (photographs taken from the top or through the bottom of a transparent container) was studied. For the color dataset, the RMSE on the percentages passing for all sieves, the coarsest sieve and the finest sieve were 2.8, 3.3, and 1.8 %, respectively. The relative error on D_{50} was 5.6 %. It should be noted that poorer performances for large particles were also observed by Manashti et al. (2021) with traditional features.

Buscombe (2020) introduced SediNet, a ConvNet model aimed at classifying sediment images and determining their PSD. SediNet includes four convolutional blocks, each consisting of between 16 and 64 convolutional filters, a batch normalization layer, and a max pooling layer. A regression model was trained to predict nine grain size percentiles including the D_{50}. The RMSE on the percentiles varied between 24 and 45 %. The predictions of SediNet were also compared to those of direct image analysis using image segmentation and texture analysis using wavelets for a small dataset of beach sand photographs by McFall et al. (2020). The mean error on D_{50} for SediNet was 22 % compared to 34 and 36 % for image segmentation and wavelet analysis.

GRAINet is another ConvNet model that was developed to predict the PSD of sediments (Lang et al. 2021). It was trained with images featuring a grain size range of 0.5 to 40 cm from 25 gravel bars along six rivers in Switzerland. GRAINet has an entry ConvNet layer of 3×3 followed by six convolutional blocks and a ConvNet layer of 1×1. The model was used to predict the PSD and the mean d_m values of the gravel bars, where $d_m = \sum P_i d_i/100$ with d_i and P_i as the mean size and percentage passing for bin i of the PSD. An RMSE of 27 % was obtained for d_m.

Both GRAINet and SediNet used a manual segmentation of each image in their dataset to determine the ground truth PSD. SediNet was trained with a dataset including 409 images of 512×512 pixels. Lang et al. (2021) trained GRAINet with a dataset including 1,491 images augmented with vertical and horizontal flipping. McFall et al. (2020) used 63 images in their comparison of SediNet with classical direct and indirect methods. The number of images in their dataset was increased to 517 by flipping the images horizontally and by using a moving window to sample different parts of each image. The datasets presented by these authors are small compared to those used to train networks for object classification. These larger datasets often include several hundred thousand images. As observed by Lang et al. (2021), networks trained with small datasets can suffer from overfitting.

Synthetic images offer one avenue to increase the size of image datasets. The dataset presented by Pirnia, Duhaime, and Manashti (2018) was more than one order of magnitude larger than the current real photographs datasets. However, it comprises highly idealized images of spherical particles. This type of dataset allows the influence of important parameters to be verified, for example the influence on RMSE of using both views from the top and bottom of a transparent container. However, the relationship between the performances obtained for synthetic images and real images remains poorly characterized (Duhaime et al. 2021).

Pretrained networks offer another approach to train ConvNets for applications where gathering large datasets is difficult. With pretrained networks, a ConvNet trained on a different dataset for a different task, often object classification with the ImageNet dataset, is applied to a different task in a different domain (Fu and Aldrich 2018). Two main methods can be used. With the first method, the network trained on a different task acts as a feature extractor. With this approach, the output of one of the fully connected layers can be used as a set of features (e.g. Fu and Aldrich 2018). With the transfer learning method, the new dataset is used to update a subset of the weights or all the weights in the pretrained model (e.g., Fu and Aldrich 2018; Peng et al. 2019). The extent of network training with the new dataset depends on the similarity of the two tasks. For similar tasks, only the last layers need to be retrained as they are associated with higher level tasks (Tajbakhsh et al. 2016). The first layers are associated with lower level tasks akin to the determination of textural features with classical methods.

Manashti et al. (2022) used pretrained ConvNets to predict the PSD from synthetic images of granular materials. They evaluated different pretrained models with transfer learning and feature extraction methods. The best results for transfer learning models were obtained with densenet201 with RMSE values on the percentages passing of 2.3, 4.1, and 3.6 % for fine, coarse, and all mesh sizes, respectively. For feature extraction, InceptionResNetV2 reached RMSE values of 2.5, 4.4, and 3.4 % on the percentages passing for fine, coarse, and all mesh sizes.

This study aims at comparing the performances of various image analysis methods to predict the PSD of granular materials from photographs. The tested methods include ConvNet model PSDNet, 15 different pretrained ConvNets with feature extraction and transfer learning methods, and nine traditional feature extractors. A dataset comprising 9600 photographs of 15 different granular materials is introduced in this article. A material removal method was used to compare the performances of each method when confronted with a new material unseen during network training. Most of the methods presented in this paper have never been applied to a real photograph dataset in the context of geotechnical engineering.

5.3 Methodology

5.3.1 Image dataset

A dataset containing thousands of images is essential to train ConvNet models like PSDNet. Three materials were selected to obtain the dataset for this project: commercial Bomix sand, 2C sand and till. The 2C sand and till were described by Dumberry, Duhaime, and Ethier (2018). A total of five materials were obtained by sieving the Bomix and 2C sands: Bomix sand retained on the 425 µm sieve, Bomix sand passing the 425 µm sieve, 2C sand retained on the 625 µm sieve, 2C sand passing the 625 µm sieve, and till. These five materials and all possible 1:1 combinations were used to generate 15 granular materials with different PSD (Figure 5.1). The PSD were obtained using sieves of 75, 150, 300, 425, 600, 630, 850, 1185, 2360, and 4750 µm.

Each material was poured in a square pan and photographed 20 times to obtain a total of 300 images for the 15 materials. The materials were mixed thoroughly in the pan between each photograph. The photographs were taken using a Nikon D90 camera with a resolution of 2848 × 4288 pixels. A nominal focal distance of 55 mm and a sensitivity of ISO 400 were used. More details on the preparation of the dataset were provided by Duhaime et al. (2021).

To increase the size of the dataset, each image was divided into 32 subimages (4 rows × 8 columns) of 512×512 pixels to create 9600 images. The 512×512 images have a scale of 56 microns per pixel. This image sampling adds some errors on the PSD. Because of segregation during mixing, each sampled region should correspond to a slightly different PSD (e.g. Dubé et al. 2021). Moreover, the subimages situated in the corners of the primary images can be slightly out of focus and blurry (e.g. photograph 1 in Figure 5.2). These sampling errors are assumed small compared to the errors associated with the PSD determination with the image analysis techniques. Figure 5.2 shows some examples of images for the 15 materials.

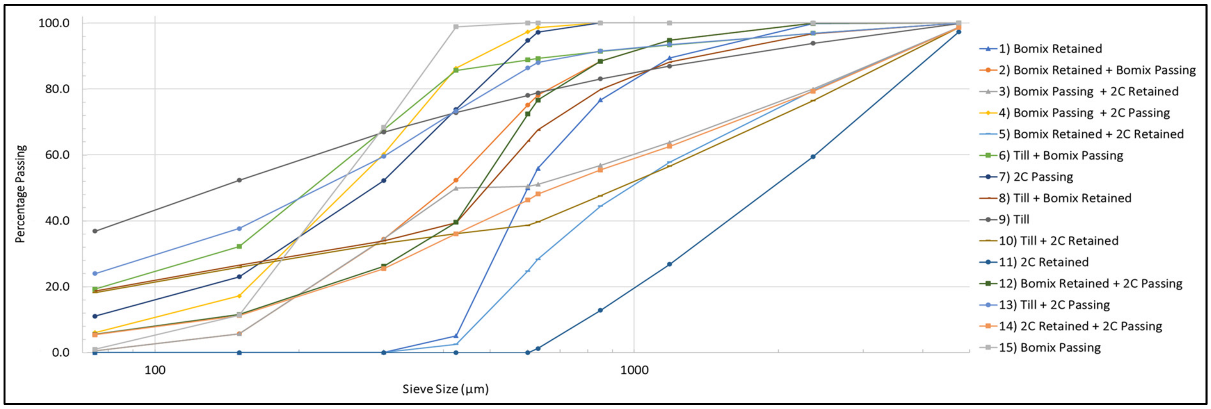

Figure 5.1 Particle size distribution of the 15 granular materials

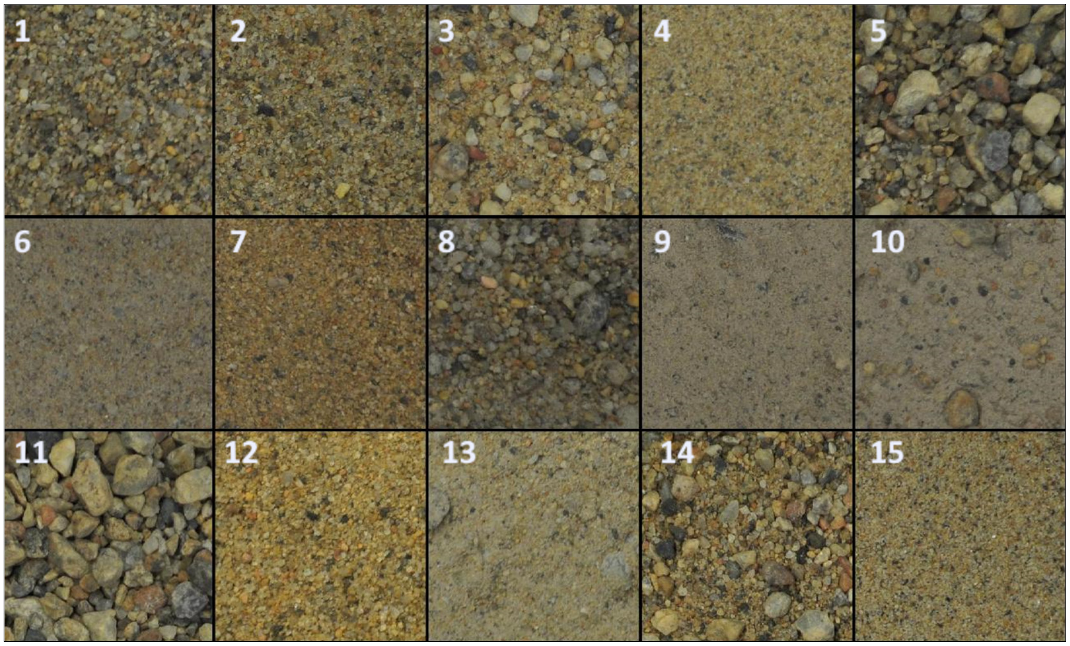

Figure 5.2 Image samples for the 15 granular materials (512×512 pixels, the numbers correspond to those presented in Figure 5.1)

5.3.2 PSDNet

PSDNet was developed by Manashti et al. (2022) to predict the PSD of granular materials from images. It was developed using a large dataset including 53 103 synthetic images obtained with the discrete element method. In this paper, the PSDNet networks developed with the synthetic image dataset were trained from scratch to predict the PSD from real photographs of granular materials.

PSDNet was also implemented as a feature extractor. The last fully connected layer before the regression layer provided 100 features for each image. These features allowed the training of a series of smaller and more computationally efficient artificial neural networks (ANN). Usage of pretrained convolutional networks as feature extractors and ANN training is introduced in more detail later in the paper.

Table 5.1 gives the structure of PSDNet for color images of 160×160 pixels. PSDNet is centered on four convolutional blocks, each consisting of one convolutional layer and three fully connected layers. The three fully connected layers include batch normalization, ReLU, and max pooling layers. A dropout layer is also added after blocks 2 and 4. The four convolutional blocks are followed by three fully connected blocks. The fully connected blocks each include fully connected, batch normalization, ReLU, and dropout layers. The last block includes a fully connected layer and regression output to predict the percentages passing for the 10 sieves used for the PSD determination. PSDNet was described in detail by Manashti et al. (2022), together with the fine-tuning process for different image sizes. In the present study, PSDNet is used without any change to train and predict grayscale and color images of 32, 64, 96, 128, and 160 pixels. Only the top view was captured and used.

PSDNet was also implemented as a feature extractor. The last fully connected layer before the regression layer provided 100 features for each image. These features allowed the training of a series of smaller and more computationally efficient artificial neural networks (ANN). Usage of pretrained convolutional networks as feature extractors and ANN training is introduced in more detail later in the paper.

Table 5.1 PSDNet layers descriptions for 160×160-pixels RGB image input

PSDNET LAYERS	DESCRIPTION
IMAGE INPUT	160×160×3 images with 'zero center' normalization
CONVOLUTION	160 5×5×3convolutions with stride [1 1] and padding [1 1]
BATCH NORMALIZATION	Batch normalization with 160 channels
RELU	ReLU
MAX POOLING	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
CONVOLUTION	320 5×5×160convolutions with stride [1 1] and padding [1 1]
BATCH NORMALIZATION	Batch normalization with 320channels
RELU	ReLU
MAX POOLING	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
DROPOUT	20% dropout
CONVOLUTION	640 5×5×320 convolutions with stride [1 1] and padding [1 1]
BATCH NORMALIZATION	Batch normalization with 640 channels
RELU	ReLU
MAX POOLING	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
CONVOLUTION	640 5×5×640 convolutions with stride [1 1] and padding [1 1]
BATCH NORMALIZATION	Batch normalization with 640 channels
RELU	ReLU
MAX POOLING	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
DROPOUT	20% dropout
FULLY CONNECTED	Fully connected layer with 1500 nodes
BATCH NORMALIZATION	Batch normalization with 1500 channels
RELU	ReLU
DROPOUT	20% dropout
FULLY CONNECTED	Fully connected layer with 500 nodes
BATCH NORMALIZATION	Batch normalization with 500 channels
RELU	ReLU
DROPOUT	20% dropout
FULLY CONNECTED	Fully connected layer with 100 nodes
BATCH NORMALIZATION	Batch normalization with 100 channels
RELU	ReLU
DROPOUT	20% dropout
FULLY CONNECTED	Fully connected layer with 10 nodes
REGRESSION OUTPUT	mean-squared-error with response' Response'

5.3.3 Traditional feature extraction

In this study, nine traditional feature extraction methods were used to generate features that were fed to an ANN. The MATLAB codes developed by Manashti et al. (2021) were used after minor modifications. Grayscale images with a resolution of 512 × 512 pixels were used to train and predict for traditional feature extraction.

5.3.3.1 Histogram of Oriented Gradients (HOG)

Histograms of oriented gradients (HOG) describe an image with a count of the different orientations of the gradient of gray level intensity for a series of predefined square or rectangular cells (Dalal & Triggs, 2005). For each cell, the gradients are allocated to a series of bins. The gradient magnitude is used to weight the gradient when it is assigned to the bins. Larger blocks with several cells are also defined to normalize the content of each bin according to the norm of all bins in the block. Normalization allows for the consideration of local contrast variations in an image. Cells that are smaller or of a similar size to the particles are usually dominated by a small number of particle edges that give the gradient a preferential orientation. More random gradient orientations will result from cells that are larger than the particles (Pang et al., 2011).

In the present study, HOG were calculated for nine orientation bins and cells of 2, 4, 8, 16, 32, and 64 pixels. The MATLAB function *extractHOGFeatures* was used. The mean and standard deviation was calculated for all HOGs to generate 12 features for the ANN inputs. The histograms were normalized with blocks of four cells with an overlap of a cell width for contiguous blocks.

5.3.3.2 Local Entropy of the grayscale image

Entropy is a statistical parameter that measures the randomness of the gray-level intensity distribution in the neighborhood of a pixel (Gonzalez & Woods, 2002). Lower local entropy values, or more constant gray levels, are expected to be associated with large particles. The nine filters presented by Manashti et al. (2021) were used to define the neighborhood around the center pixel. The MATLAB functions *Getnhood* and *Strel* were used to set these filters. Entropy was calculated with the *entropyfilt* function. Entropy features were calculated for bin sizes of 4, 8, 16, 32, 64, 128, and 256 pixels. The mean and standard deviation of the entropy for the nine disk filters and the seven bin sizes were computed and applied as the ANN inputs (total of 126 features).

5.3.3.3 Local Binary Pattern (LBP)

With local binary patterns (LBP), the pixels of a neighborhood are compared to a threshold set by the centre pixel (Ojala, Pietikainen, and Maenpaa 2002). The pixels in the neighborhood take values of 0 or 1 depending if their gray level is higher or lower than the threshold value (Sudha & Ramakrishna, 2017). This pattern is coded in a binary number of *p* bits, where *p* is the number of pixels in the neighborhood around the center pixel. Different numbers of neighboring pixels and distances between the center and the neighboring pixels may be selected (Kaya, Ertuğrul, and Tekin 2015). The binary numbers that only differ by a rotation can be combined into bins to reach rotation invariance, and thus limit the number of features. The MATLAB functions *extractLBPFeatures* was utilized to calculate LBP. In this study, LBP was computed for 20 neighbors at a radius of 3 pixels for a total of 383 features. These LBP features were used as ANN inputs.

5.3.3.4 Local configuration pattern (LCP)

Local Configuration Patterns (LCP) combine LBP with Microscopic configuration of images (MiC) (Deng and Yu 2015; Guo, Zhao, and Pietikäinen 2011). LBP does not describe the variability of the gray level intensity. MiC provides this information by finding the linear combination of neighboring pixel intensity that best describes the intensity of the center pixel. This linear combination is defined for each rotation invariant LBP bin. A least square approach is used to calculate the weights of the linear combinations. The LCP features that are fed to the

ANN are a combination of the LBP histogram and MiC. In this study, the LCP features were calculated for 20 neighboring pixels at a radius of 3 pixels, for a total of 441 features. The calculations were made with the MATLAB script described by Manashti et al. (2021). This scripts uses the original LCP code developed by Guo, Zhao, and Pietikäinen (2011).

5.3.3.5 Completed Local Binary Pattern (CLBP)

The Completed Local Binary Pattern (CLBP) has three parts (Guo, Zhang, and Zhang 2010). The first part corresponds to the LBP. The second part gives information on the absolute value of the gray level differences between the center pixel and the neighboring pixels. A binary number is obtained from the differences by thresholding them with the mean absolute difference for the whole image. The third part describes the center pixel intensity. It is thresholded with the mean pixel intensity to obtain another binary value. The binary numbers are then combined and used to build a histogram. The CLBP was computed for 20 neighbors at a radius of 3 pixels, similarly to LBP and LCP. The total number of features for CLBP is 44. The MATLAB script presented by Manashti et al. (2021) was used to calculate the CLBP features. It is centred on the original CLBP code of Guo, Zhao, and Pietikäinen (2012).

5.3.3.6 Fourier Transform

Image features can be calculated based on the mean and standard deviation of the magnitude spectrum of 2D Fourier transforms for a series of ring filters (Szeliski, 2011; Yaghoobi et al., 2019). Ten rings filters were used. The thickness of successive rings was increased by a factor 1.6. More details on the Fourier transform and ring filters are presented by Manashti et al. (2021). The mean and standard deviation of the magnitude spectrum of the Fourier transform allowed 20 features to be defined for the ANN inputs.

5.3.3.7 Gabor filters

Gabor filters with different wavelengths and orientations were applied to the images. Gabor filters combine the Gaussian kernel function with a sinusoid wave (Tuceryan & Jain, 1993; Yaghoobi et al., 2019), enabling the local frequency content of an image to be evaluated. The MATLAB Gabor function was used for the preparation of a filter bank corresponding to wavelengths of 3 to 15 pixels, and orientations between 0 and 150°. The mean and standard deviation of 42 filters (84 parameters) were used as ANN inputs. The MATLAB function *imgaborfilt* was used to apply the filter bank to the images.

5.3.3.8 Haar wavelet transforms

Hryciw et al. (2015) presented detailed calculations of two-dimensional Haar wavelet transforms in the context of PSD determination. Each level of the wavelet transform reduces the image size by a factor of 2. For each level and for each 4×4 pixel neighborhood in the image, four coefficients are calculated using the mean pixel intensity (A), and pixel intensity differences between rows (H), columns (V) and diagonals (D). The sum of the square of the A, H, V, and D coefficients for the whole image describes the image energy. The total energy is preserved for each level of Haar wavelet transform for a given image (Hryciw et al. 2015). To define a series of features, the sum and standard deviation of H^2, V^2, and D^2 were calculated for 7 levels of wavelet transform. This results in 42 features. The coefficients were calculated using a MATLAB code developed by Manashti et al. (2021) and centred on the MATLAB function *haart2*.

5.3.4 Pretrained ConvNet

Due to the large number of weights in their structure, ConvNets require more resources to train compared with other types of neural networks. When trained from scratch, the initial weights are usually generated using random numbers with a normal distribution, a mean of 0 and a small standard deviation. In these circumstances, pre-trained ConvNet models that have been trained on similar datasets, or even very different datasets in our case, can be adapted to save time and computing resources.

In this article, 15 ConvNet models were evaluated for transfer learning and feature extraction. The tested models include AlexNet (Krizhevsky, Sutskever, and Hinton 2012), VGG16 (Simonyan & Zisserman, 2014), VGG19 (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), InceptionV3 (Szegedy et al. 2016b), DenseNet201 (Huang et al., 2017), MobileNetV2 (Sandler et al., 2018), ResNet18 (Wu, Zhong, and Liu 2018), ResNet50 (He et al., 2016), ResNet101 (He et al., 2016), Xception (Chollet, 2016), InceptionResNetV2 (Ioffe and Szegedy 2015; Szegedy et al. 2016a), ShuffleNet (Zhang et al., 2017), NasNetMobile (Zoph et al., 2018), and NasNetLarge (Zoph et al., 2018). Depending on the structure of the model, four different sizes of images (224, 227, 299, and 331) were used for training and validation of pretrained convnets. Three methods were used for each model.

With the first method, the final SoftMax (classification layer) of the 15 networks was replaced by a regression layer to predict the percentages passing for the 10 sieve sizes. Starting from the pretrained weights, each model was trained for five epochs. Because of the fundamental difference between the tasks considered in this paper (PSD determination) and the original task with the ImageNet dataset (object classification), the weights were updated for all layers. This first method will be referred to as “transfer learning”.

With the second method, the pretrained models that were retrained using the material photograph dataset with the first method were used as feature extractors. A total of 1,000 features were extracted from the last fully connected layer. The PSD were predicted using an ANN created with the *fitnet* function in MATLAB to calculate the percentages passing for the 10 sieve sizes. The second method will be referred to as “feature extraction of transfer learning”.

With the third method, the last fully connected layer of each pretrained model was used to extract 1000 features without retraining the pretrained ConvNets with the granular material photographs. The main difference between the second and third methods is the absence of ConvNet training with the material photographs for the third method. As with the second method, the features were fed into an ANN created with the *fitnet* function in MATLAB. This method will be referred to as “feature extraction”.

It is expected that feature extraction of transfer learning will lead to better results than pure feature extraction because of the ConvNet training. The feature extraction of transfer learning allows for a minimal retraining of the ConvNet layers that accomplish lower level tasks, and a more extensive training of the fully connected layers in the ANN that execute higher level tasks. The ANN training with feature extraction methods is much faster than the ConvNet training.

5.3.5 Training PARAMETERS

The training of PSDNet, the pretrained models, and the feature extraction ANN was conducted using MATLAB 2020b. For each method, the network performances were calculated based on the RMSE of the percentages passing for all sieves. The following sections give the training parameters for each network.

5.3.5.1 PSDNet

The same parameters as Manashti et al. (2022) were used to train PSDNet. The initial learning rate was set to 0.0001 and the number of epochs was set to 60. Weights were updated with the Stochastic Gradient Descent with Momentum (SGDM). The training data was shuffled after every epoch during training. Predictions were made for 10 sieve sizes (75, 150, 300, 425, 600, 630, 850, 1185, 2360, 4750 µm). Training was conducted using two graphical cards (Nvidia GeForce RTX 2080 and GTX 1650). Compared with Manashti et al. (2022), one significant difference is the shorter time needed to train PSDNet because of the lower number of images (9600 rather than 53003).

Network performances are determined using a test dataset. This subdataset it not shown to the neural network during training and validation. Two types of test datasets were used. With the first method, the test dataset comprises 10 % of the images and is randomly sampled. This will be referred to as the random testing dataset in the paper. The 15 materials are included in the training, testing, and validation datasets with this approach. 7680 photographs (80%) were randomly selected as the training dataset, 960 images (10%) as validation dataset, and the same number as the test dataset. The second method will be referred to as the material removal test. It was designed to verify the performances of ANN and ConvNets if images of a new material, unseen during training, are fed to the networks. With this method, the 15 granular materials were successively removed from the training and validation datasets and used as the testing dataset. The ability of each method to predict the PSD of unseen materials is expected to give a more realistic portrait of their performances. With the material removal test, the test dataset contains 640 images of a given material. The 14 remaining materials were sampled randomly to fill the training (8320 images) and validation datasets (640 images). The material removal technique was inspired by the geographical cross-validation used by Lang et al. (2021).

5.3.5.2 Pretrained models

Pretrained models followed PSDNet parameters and specifications for the training, validation and test process. However, in pretrained models, five training epochs were performed, and the initial learning rate of 0.000001 was applied. The size of the images varied according to the pretrained model (224, 227, 299, and 331 pixels). Training, validation and testing was conducted using both random testing datasets and the material removal test.

5.3.5.3 ANN for traditional features and feature extraction

Function fitting neural networks (*fitnet*) were used in MATLAB 2020b to predict the percentage passing for the 10 sieves using extracted features. The hyperbolic tangent sigmoid transfer function was used as with the ANN models presented by Manashti et al. (2021).

Network training was conducted with the scaled conjugate gradient method. The maximum number of training epochs was set to 10,000 for all networks.

The numbers of layers, and neurons on each layer were determined by trial and error. They vary with the number of inputs, which ranged from 12 for HOG up to 1208 for a combination of all traditional features. For example, a four-layer network with 20, 13, 17, and 10 neurons on each layer was used with HOG for 12 features, whereas three layers with 200, 100, and 50 neurons were used for LBP with 383 features. Networks with a single layer with ten neurons were utilized for pretrained feature extraction, and transfer learning feature extraction methods with 1000 features.

The testing, validation and training subdatasets were chosen using the same approach as for the training of PSDNet. For the random testing dataset, the 9600 images were divided randomly between training (70 % of images), validation (15 %) and testing (15 %) subdatasets. The soil removal test used the same subdataset proportions as with PSDNet. One material out of 15 was removed from the training and validation subdatasets and used as the testing subdataset (640 images). The remaining images were randomly assigned to the testing (8320 images) and validation (640 images) subdatasets.

5.4 Results

Results for each method are presented in this section. The RMSE on the percentages passing for each sieve is used as the main performance metric.

5.4.1 PSDNET

PSDNet was assessed using grayscale and color images with sizes of 32, 64, 96, 128, and 160 pixels. Based on Figure 5.3, the best results were achieved for the color images with a size of 160 pixels with an RMSE value of 3.7 %. For the grayscale dataset, the best results were obtained with 128-pixel images with an RMSE of 5.5 %. For a given image size, color images led to better results with RMSE of 3.7 to 7.3 %, compared with RMSE of 5.5 to 8.4 % for grayscale images. On the other hand, for a given image size, the grayscale images required less computing resources for the ConvNet training.

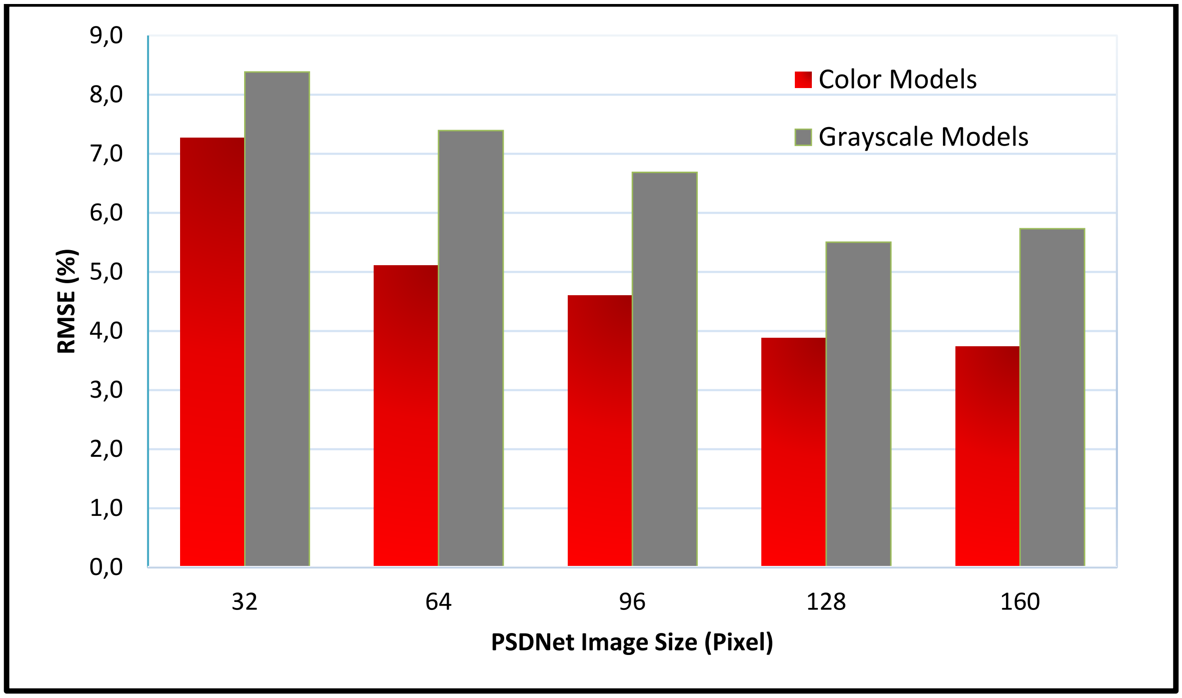

Figure 5.3 Results of running PSDNet on grayscale and color images of different sizes

The results for the material removal test for grayscale and color images with PSDNet are presented in Figure 5.4. The best results were obtained for material 4 (passing Bomix and 2C passing) with mean RMSE of 2.8 and 3.9 % for grayscale and color images, respectively. It should be noticed that the best results were obtained for fine-grained materials with similar PSD (materials 4, 6, 7, and 15, Figure 5.1). Several explanations can be suggested for these results, such as the presence of a larger number of similar PSD in the training dataset and less segregation in the fine-grained materials during sample preparation for the photographs. Better performances for finer sieves have also been noted before (e.g. Manashti et al. 2021). The worst RMSE were obtained for material 9 (till) with an RMSE value of 18.5 % for the color dataset. Materials 9, 10 and 11 had the highest RMSE values. Materials 9 and 11 correspond respectively to the finest and coarsest materials. Material 10 has a relatively wide PSD compared to the other materials. This tend to support the hypothesis that better results are obtained with the material removal test when the training dataset includes different materials with similar PSD.

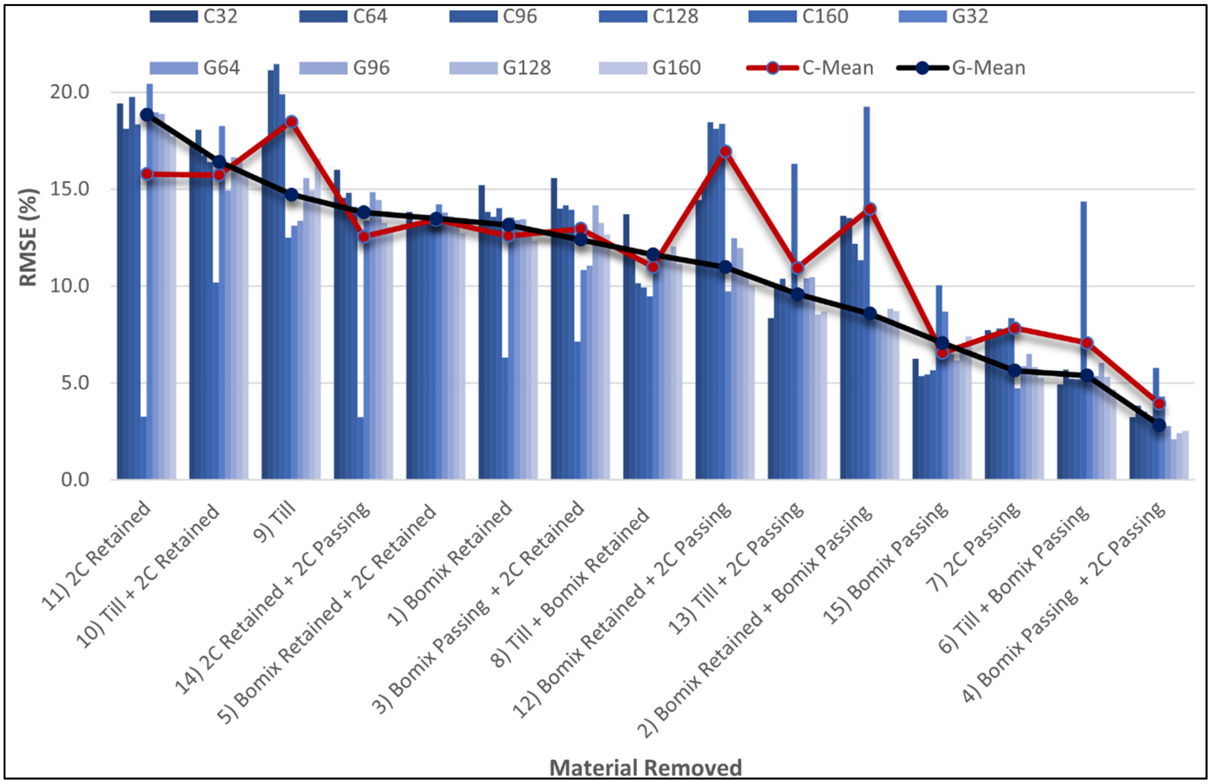

Figure 5.4 Results for the material removal test of color and grayscale PSDNet for different image sizes

5.4.2 PSDNET used as feature extractor

Combining extracted features for various image sizes can improve the prediction accuracy. For this reason, features extracted from the last fully connected layer of the PSDNet networks trained with different images sizes were used as ANN inputs. In this study, 100 features were extracted from the last fully connected layer of PSDNet for both color and grayscale images of 32, 64, 96, 128, and 160 pixels, for a total of 1000 available features. Figure 5.5 shows the results of feeding different feature combinations to the ANN for color and grayscale images.

Using a combination of extracted features for grayscale images can predict the PSD with an RMSE of 3.2 %. Without feature extraction, PSDNet had RMSE values between 5.5 and 8.4 % for grayscale images. For color images, a combination of all PSDNet features led to an RMSE of 2.3 %. In comparison, the RMSE for PSDNet was between 3.7 and 7.3 %. Combining all PSDNet grayscale and color features (1000 features) led to the best results, with an RMSE value of 1.8 %.

The combination of PSDNet features led to good results in the material removal test. As shown in Figure 5.6, combining all PSDNet gray and color features (1000 features) led to mean RMSE values of 3.3 and 18.4 % for material 4 and 11, the best and worst results, respectively. The mean RMSE for the combination of gray, color and both gray and color PSDNet features were 12.2, 10.6, and 9.1 %, respectively. Figure 5.7 presents the real and predicted PSD for the material removal test with randomly selected images for each of the 15 materials. The subfigure numbers correspond to the PSD in Figure 5.1. Predictions were made by combining the features extracted from PSDNet for all image sizes (32 to 160 pixels), for gray and color images, for a total of 1000 features. Material 12 had the lowest error with an RMSE of 2.7 %. Conversely, in the worst case, the PSD of material 1 was predicted with an RMSE value of 22.1 %. It should be noted that the error for material 1 corresponds approximately to the mean RMSE for the materials with the worst performances in the material removal test for PSDNet (figs 4 and 6).

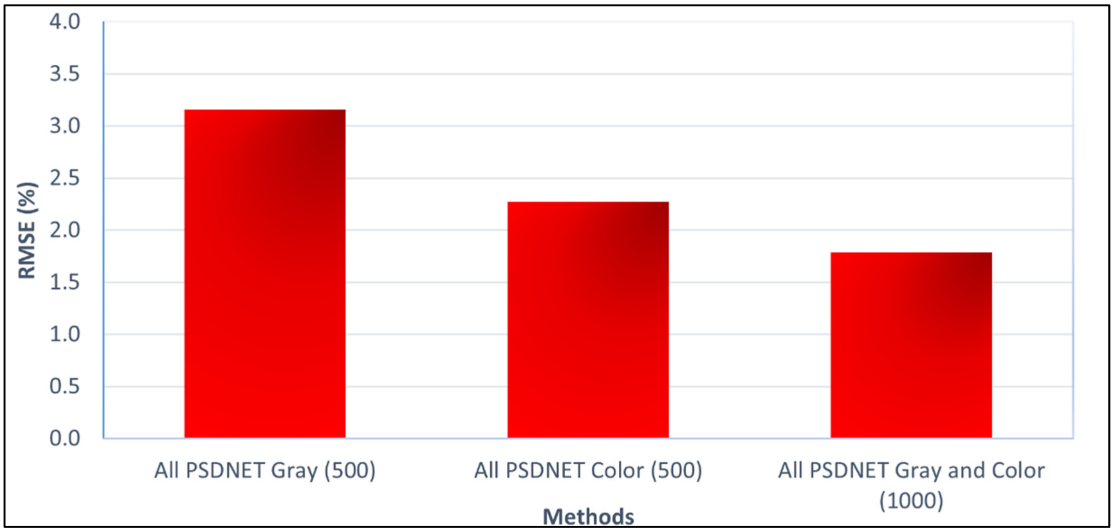

Figure 5.5 Results of combining extracted features of different color and grayscale PSDNet

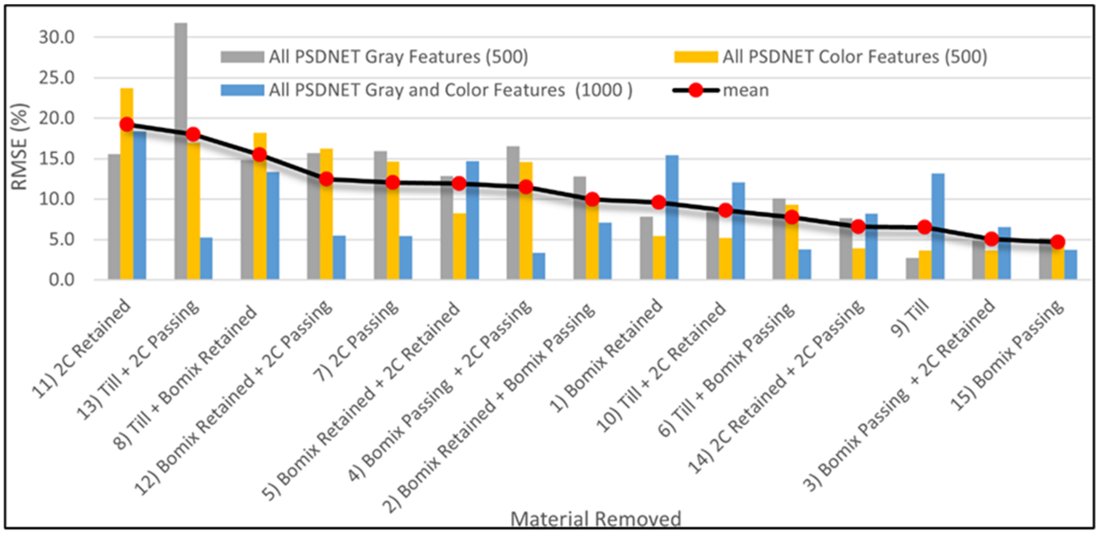

Figure 5.6 Results for combination of gray, color, and both gray and color PSDNet extracted features to predict the PSD in the material removal test

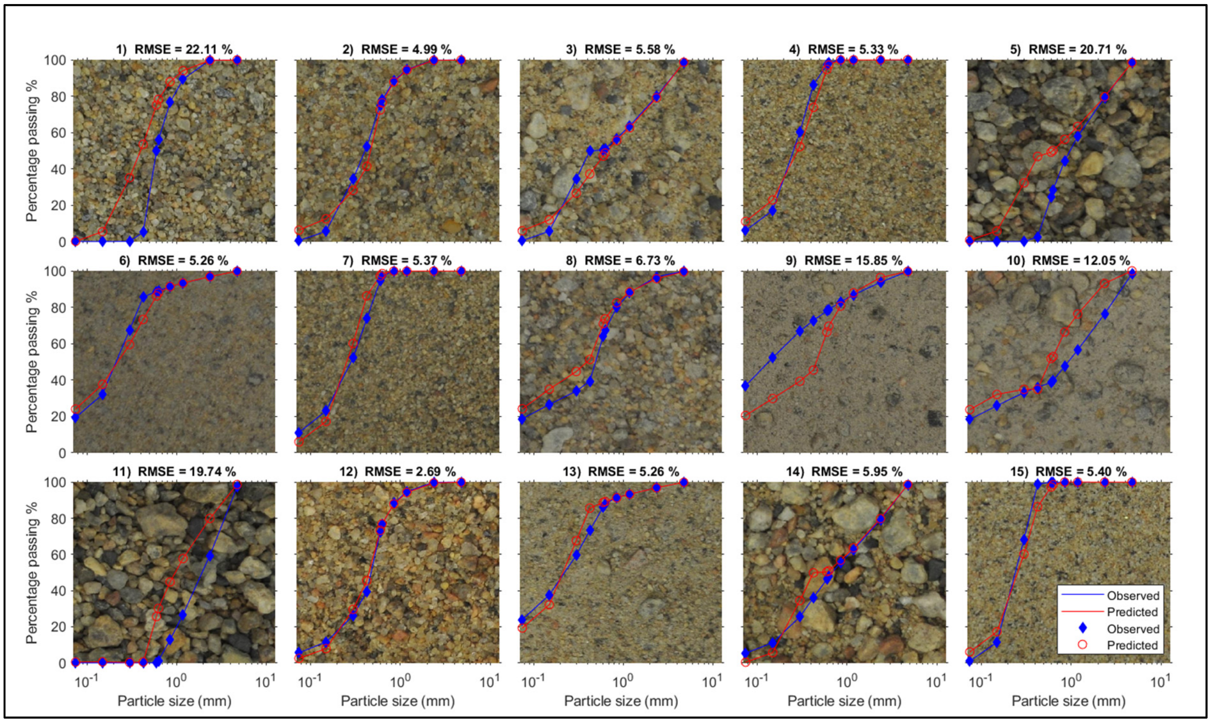

Figure 5.7 PSD comparison for randomly selected images for the material removal test. The blue curves were obtained by sieving and the red curves correspond to the network predictions obtained with PSDNet used as a feature extractor for all image sizes and for both grayscale and color images

5.4.3 Pretrained models

Figure 5.8 compares the RMSE on the percentages passing for the 15 pretrained models. The three methods that were previously introduced are compared (transfer learning, feature extraction of transfer learning, feature extraction). The best results were obtained with transfer learning followed by feature extraction. The best results were achieved with InceptionResNetV2 (RMSE of 1.7 %) followed by NasNetLarge and ResNet101 (RMSE of 2.4 %). GoogLeNet, with an RMSE value of 5.6 %, had the worst performances for feature extraction of transfer learning. Transfer learning of pretrained models led to RMSE values between 4.3% for NasNetLarge and 9.3 % with AlexNet. Feature extraction of pretrained ConvNets achieved RMSE values from 4.0 % with ResNet101 to 6.2 % with GoogLeNet and AlexNet.

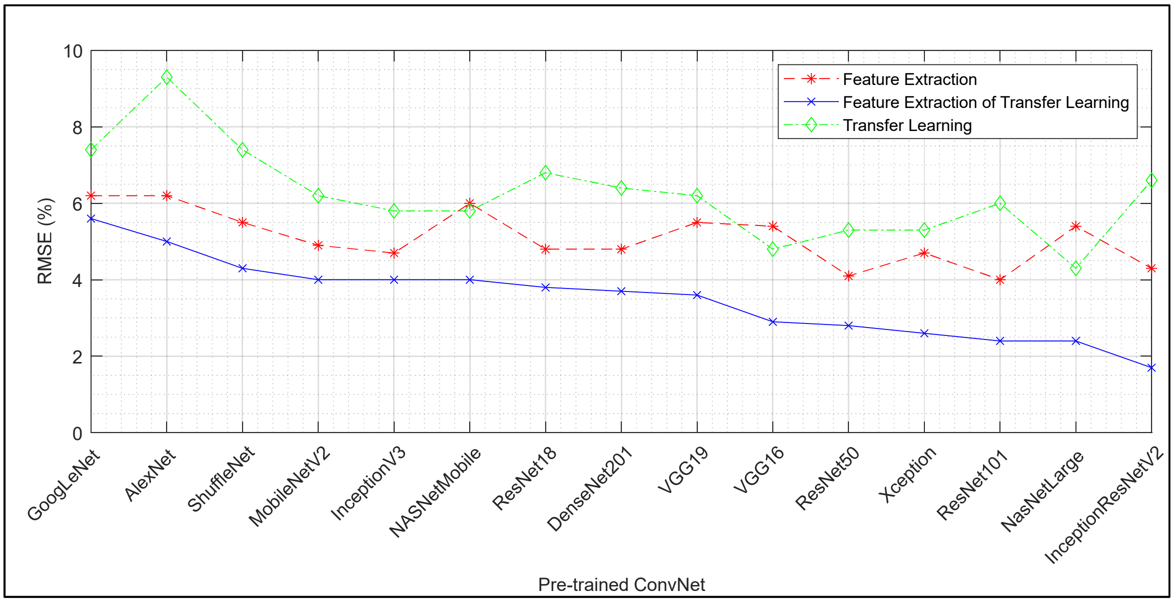

Figure 5.8 Evaluating of pretrained ConvNet in three-position, including transfer learning, feature extraction, and feature extraction of transfer learning for PSD prediction

Results for the material removal test for transfer learning models are shown in Figure 5.9. As expected, the RMSE increases in contrast with the random testing dataset. Material 4 (Passing Bomix and 2C Passing) led to the best performances with a mean RMSE for all pretrained networks of 6.1 %, whereas material 11 (2C Retained) led to the poorest performances with an RMSE of 17.0 %. With regards to the performances of each network, Xception and ResNet101, with mean RMSE of 9.3 and 17.3 %, had the best and worst performances, respectively. GoogLeNet could predict the percentages passing for material 4 with a minimum RMSE of 1.5 %, while ResNet101 reached an RMSE of 30.2 % for material 11.

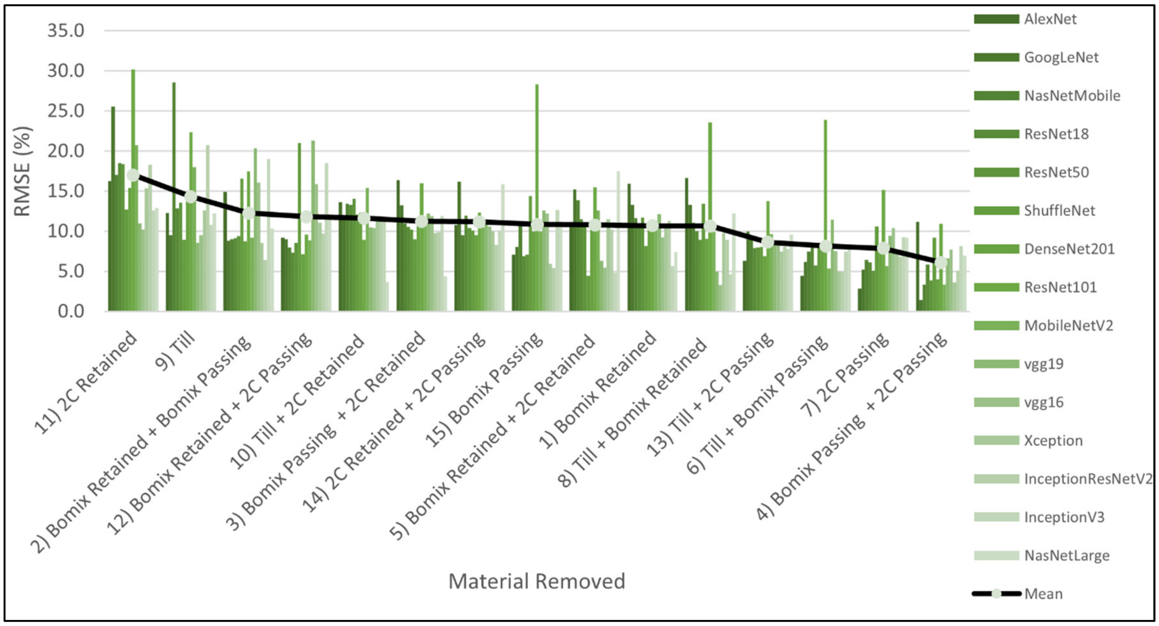

Figure 5.9 Results of evaluating pretrained model as transfer learning for the material-removal test

5.4.4 Traditional Feature Extraction

Results for traditional feature extractions are presented in Figure 5.10. A combination of all methods with 1208 features reached the best results with an RMSE on the percentages passing of 2.7%, followed by entropy, CLBP, and LBP with RMSE of 3.1, 3.3, and 4.0 %, respectively. As with other techniques, the RMSE increases for the material removal test (Figure 5.11). The best mean RMSE was obtained for material 4.3 %. Material 10 led to the poorest performances with an RMSE of 17.9 %. With respect to the methods, wavelet and a combination of all methods were the top models with a mean RMSE of 3.2 % for material 4. Entropy led to the highest prediction error with a mean RMSE value of 24.7 % for material 10.

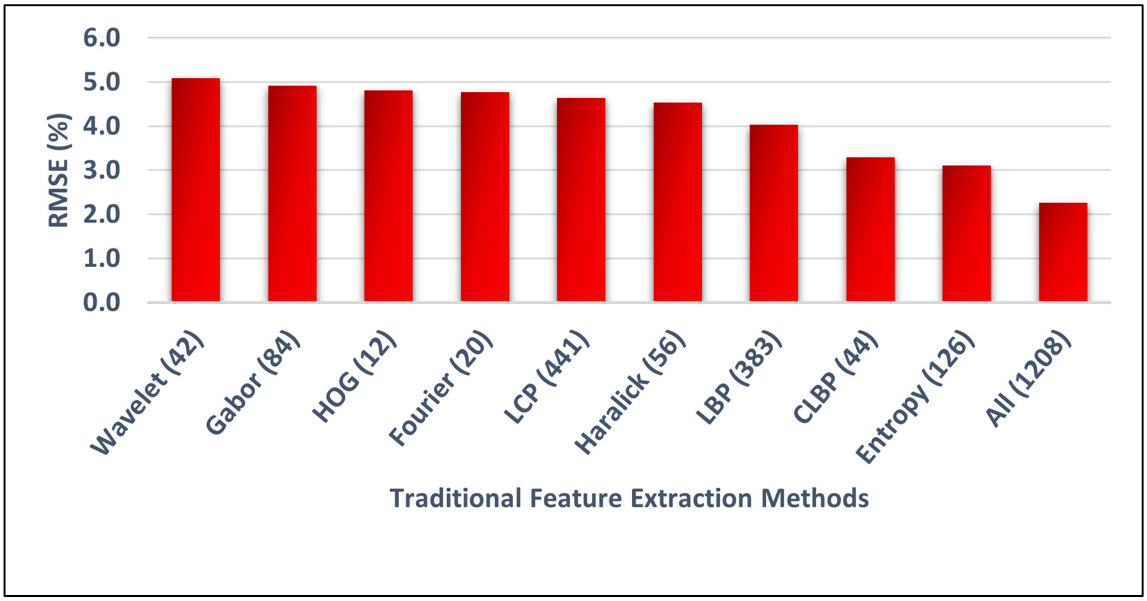

Figure 5.10 RMSE on percentages passing for traditional feature extraction methods (the numbers in parentheses following the method name corresponds to the number of features)

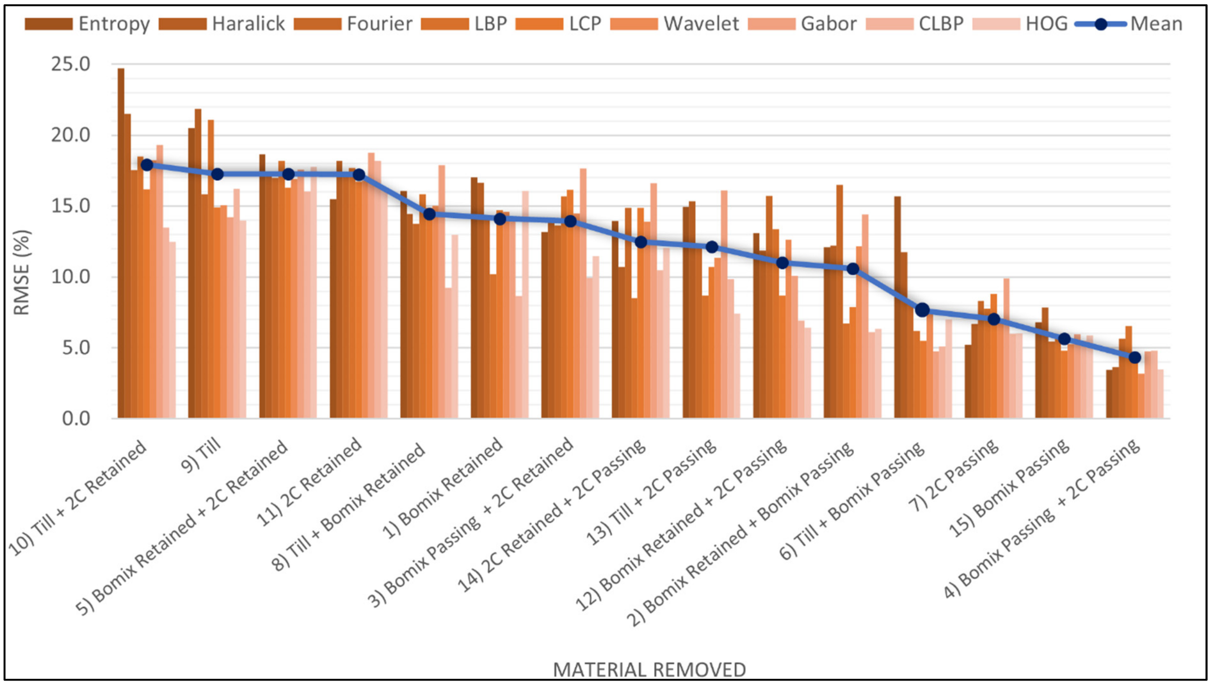

Figure 5.11 RMSE for the material removal test with traditional feature extraction methods

5.4.5 D50

Error percentages for the *D_{50}* predicted with PSDNet for random testing of color and grayscale images are shown in Figure 5.12. The error generally decreases with the images size. For the color dataset, the best performances were obtained with the 160-pixel images with a percentage error on *D_{50}* of 7.9 %. For grayscale images, the lowest error on *D_{50}* was obtained with the 128 pixel images with an error of 11.2 %

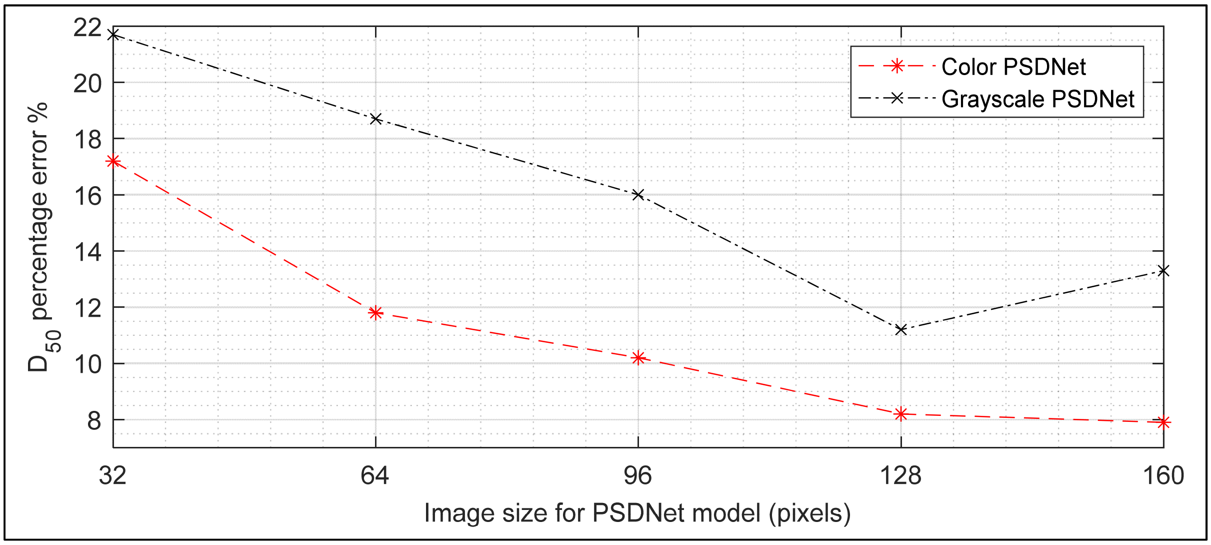

Figure 5.12 Percentage error on *D_{50}* for various sizes of PSDNet color and grayscale dataset

Percentage error on *D_{50}* for different pretrained ConvNets used with the transfer learning method are shown in Figure 5.13. VGG16, with an error of 11.3%, followed by NasNetLarge and ResNet50 by an error of 12.8 %, had the best performances. AlexNet had the worst performances with a percentage error on *D_{50}* of 26.0 %.

Combining all 1208 traditional features led to a *D_{50}* percentage error of 4.4 %. Features extracted from PSDNet for all color and grayscale images (1000 features) led to a *D_{50}* percentage error of 3.4 %. A slightly higher errors were obtained with features for color images (500 features, 4.8 %) and gray scale images (500 features, 4.9 %).

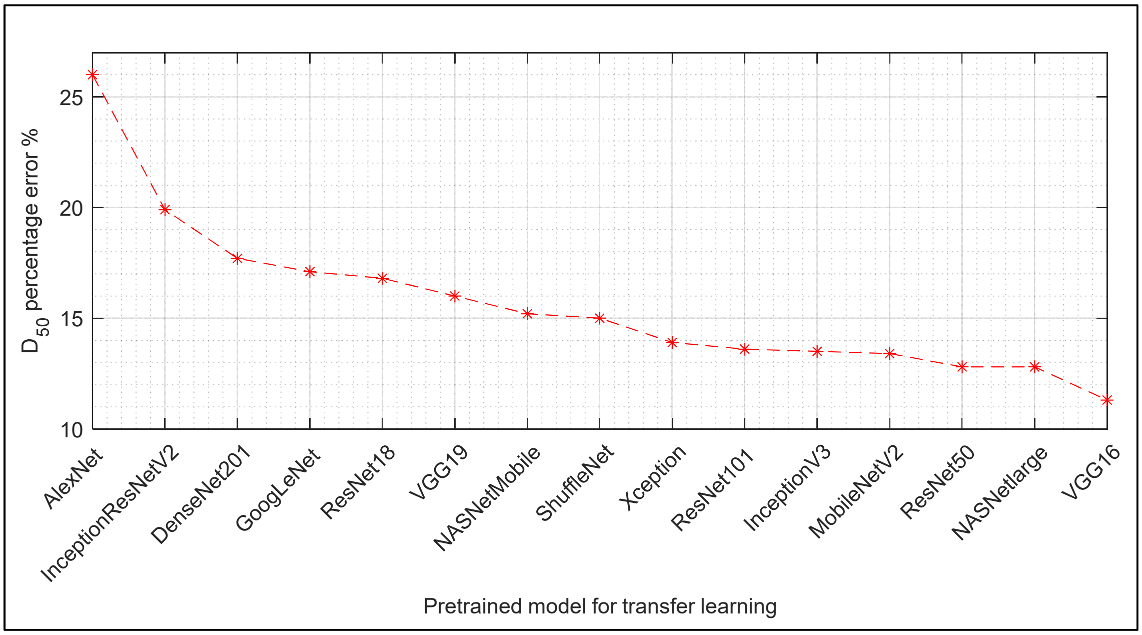

Figure 5.13 Relative error on *D50* for transfer learning with pretrained models

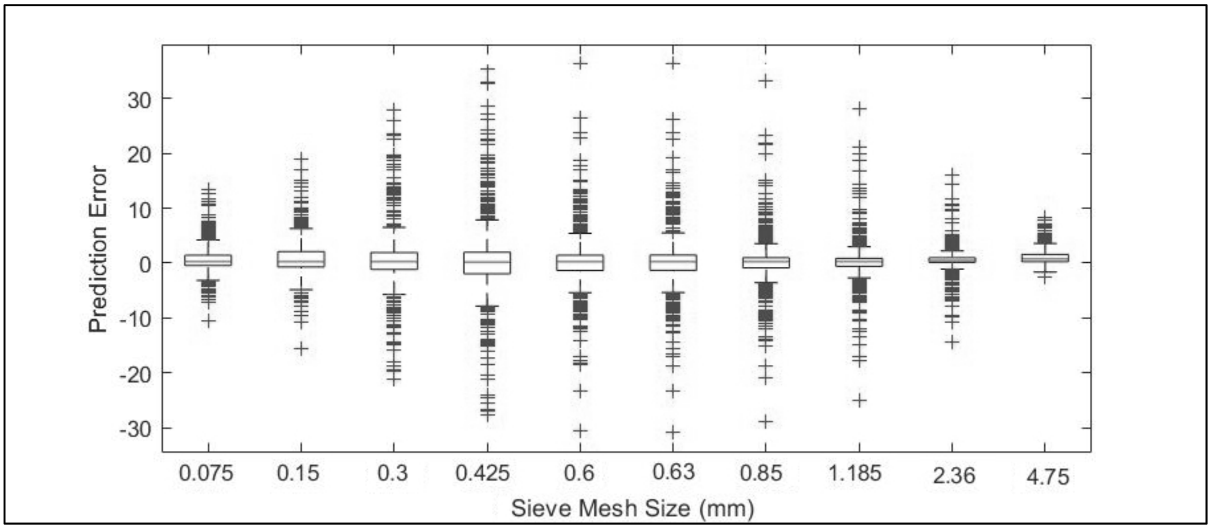

Figure 5.14 Prediction error for different sieve sizes for PSDNet with color images of 160 pixels

5.4.6 Prediction error chart

Figure 5.14 shows the prediction error for PSDNet in terms of percentages passing for the different sieve sizes for color images of 160 pixel. The prediction error is determined based on the real percentage passing minus the predicted percentage passing for each mesh size. The absolute value of the error is lower for the coarsest sieve (4.75 mm) because the range of percentages passing for this sieve in the dataset is narrow. The largest range of error was obtained for the 0.6 and 0.63 mm sieves. The prediction errors for these sieves range from -31 to 36 %.

5.5 Discussion

The mean RMSE on the percentages passing obtained in this study for real photographs of granular materials can be compared to the results presented by Manashti et al. (2021; 2022) for synthetic images. In this study, PSDNet yielded RMSE on percentages passing ranging from 3.7 to 8.4 % for real photographs. The RMSE ranged from 1.7 to 9.3 % for pretrained models, and from 2.3 to 5.1 % for traditional feature extraction. In comparison, for synthetic images of the top surface of particle assemblages in a container, Manashti et al. (2022) obtained RMSE values on the percentages passing ranging between 4.2 and 7.2 % for PSDNet. For pretrained models, the RMSE ranged from 4.6 to 7.5 %. For traditional feature extraction with the same dataset, Manashti et al. (2021) obtained RMSE values ranging between 3.4 and 6.9 %.

Therefore, the results obtained for both types of images are generally comparable. Similar conclusion could be drawn from a comparison of the D_{50} errors. It should be noted that the range of PSD that were considered in this study and for the synthetic images used by Manashti et al. (2021; 2022) are different. Manashti et al. (2021; 2022) considered a smaller range of particle size (from 0.075 mm to 1.18 mm, versus 0.075 mm to 4.75 mm in this study), but a much larger number of distinct PSD (53003 versus 15 in this study). A comparison was conducted for very similar range of PSD distributions using local entropy features as ANN inputs by Duhaime et al. (2021). Very similar RMSE were obtained for the real photographs and the synthetic images with RMSE of 3.8 and 3.4 %, respectively.

Using synthetic images for network training can have several advantages. First, it is much easier to generate large datasets of synthetic images than to obtain real photographs and their matching ground truth. For PSD determination, gathering real photographs of granular materials with their PSD is time consuming. Synthetic images can also be used to test the influence on network performances of different dataset parameters (e.g., lighting, camera position). A good example is the comparison of PSDNet performances by Manashti et al. (2022) for synthetic datasets including both views from the top and bottom of a transparent container with performances for datasets including only views from the top. Lower RMSE on the percentages passing can be obtained when both viewpoints are considered (Manashti et al. 2022). Duhaime et al. (2021) have also shown that real and synthetic images can be combined in the same dataset for network training with RMSE that are comparable to those obtained with only synthetic or real images. For other applications, it has been demonstrated that combining real and synthetic images for the training of ConvNets or ANN can increase the generalization potential of networks and avoid overfitting (Nikolenko 2021).

The RMSE values that were obtained in this study can also be compared with published results for real photographs analyzed with other methods, such as direct methods based on image segmentation. RMSE values from the material removal test are probably the most representative performance metric for real applications where the tested material will always be an unseen material. The mean RMSE values that were obtained for the material removal test in this study varied between 2.8 and 19.2 % depending on the method and the material. With direct methods and real photographs, Liu and Tran (1996), and Sudhakar, Adhikari, and Gupta (2006) reached RMSE values on percentages passing ranging between 13 and 36 %, and 15 and 20 %, respectively. In comparison, better performances were obtained in this paper for the best methods with materials with PSD at the center of the grading range in Figure 5.1 (e.g. material 7). The performances for materials with extreme PSD (e.g. material 11) led to performances that are comparable with previously publish results.

Our results suggest that some methods are more computationally efficient than others. The RMSE values obtained for each method are similar, but the computational resources that are required are very different. Traditional feature extraction and pretrained networks used as feature extractors proved to be the most efficient methods. The difference in computational resources is due to the much smaller ANN networks required with feature extraction methods compared to ConvNets, and to the much smaller input size of feature vectors compared to images.

5.6 Conclusion

The primary purpose of this paper was to assess the performances of PSDNet to predict the PSD for 9600 photographs of 15 different granular materials. Grayscale and color images of different sizes were tested to predict the percentages passing for 10 sieve sizes. Several others methods were used to predict the PSD, including traditional feature extraction, pretrained ConvNet with transfer learning, feature extraction, and feature extraction of transfer learning. The majority of these methods had been evaluated previously with synthetic datasets by Manashti et al. (2021; 2022).

Each method could predict the PSD accurately. PSDNet could predict the percentages passing with RMSE values of 3.7 and 5.5 % for color and grayscale images, respectively. Feature extraction of transfer learning for the pretrained network InceptionResNetV2 and a combination of all PSDNet gray and color features led to the best results with RMSE values of 1.7 and 1.8 %, respectively.

A material removal test was used to determine the performances of each method when confronted with an unseen material for real applications. All methods could predict the PSD of unseen materials with RMSE values on the percentages passing ranging between 2.1 and 31.8 %. Features extracted from PSDNET for grayscale and color images (1000 features) led to the lowest RMSE (9.1 %), while local entropy led to the highest RMSE (14.1 %). All methods that were presented in this paper were able to predict the PSD of unseen material with performances that were similar, or in some case better, than previously published results obtained with classical methods like segmentation.

5.7 Acknowledgments

The authors gratefully acknowledge the funding of Hydro-Québec and the National Sciences and Engineering Research Council of Canada (NSERC) for this project.

CHAPTER 6

CONCLUSION AND RECOMMANDATION

6.1 CONCLUSION

This thesis aimed to find an accurate technique for predicting PSD using image processing techniques. To this end, various techniques were examined, including traditional feature extraction, ConvNet, and pretrained ConvNet. The significant findings drawn from this study are summarized in this section.

This thesis utilized two datasets comprising synthetic and real images of granular materials. The first dataset contained 53003 pairs of synthetic granular material images. The synthetic dataset was utilized for training ConvNet and ANN to predict the PSD. The synthetic dataset included the percentages passing for 5 mesh sizes. Synthetic image datasets give many advantages. Since they are simpler to produce than real image datasets, the influence of various parameters can be easily checked. For example, the effect of lighting on the efficiency of various techniques could be examined by applying a randomly placed light source prior to the image rendering. Synthetic images could also be utilized to increase the size of real image datasets. The second dataset contained 9600 real photographs of 15 granular materials. Percentages passing for 10 mesh sizes were available for this dataset. For determining particle sizing of the soil sample, Sieving was chosen as standard ground truth for geotechnical engineering.

The influence of the image size and color was verified for both real and synthetic datasets. Image sizes of 32, 64, 96, 128, and 160 pixels were examined on the grayscale and color bands. Better results were obtained for the larger color images.

A series of traditional textural features were utilized as inputs for ANN. Good results were obtained for the PSD prediction, whether using features individually or a combination of them. Techniques based on local binary patterns (LBP), such as CLBP and LCP, could better predict the PSD than the other feature extraction techniques. To the best of our knowledge, this study provided the first example of PSD estimation using these techniques. The combination of features achieved the best results for the real and synthetic datasets. For the synthetic dataset, the RMSE value on the percentages passing for the coarsest, finest, and all sieves were 4.7, 1.7, and 3.4 %, respectively. The relative error on D_{50} was 6.1 %.

To the best of our knowledge, the first application of ConvNets for the estimation of PSD in geotechnical engineering was made by Pirnia et al. (2018) as part of this thesis. This thesis demonstrated that ConvNets could form the basis of a new type of fast, reliable, and cost-effective online PSD prediction methods in the laboratory or in the field. Compared to traditional feature extraction, better results were obtained for coarse particles with PSDNet (model C160STU). For the synthetic dataset, the RMSE on the percentages passing for the coarsest sieve were 3.4 and 4.7 % for PSDNet (model C160STU) and the combined traditional features, respectively. For fine particles, almost identical results were obtained for synthetic images using PSDNet (model C160STU) and combined traditional features, with RSME values of 1.7% and 1.8%, respectively. For all sieve sizes, PSDNet (model C160STU) and the combined feature extraction technique reached RMSE values of 2.8 and 3.4% for the synthetic dataset.

To our knowledge, this is the first time that pre-trained ConvNets have been utilized as feature extraction or transfer learning method in geotechnical engineering. A selection of pretrained models was evaluated using synthetic granular soil and real soil images, leading to excellent results. Results from feature extraction of pretrained ConvNet were almost the same as those of the combined traditional feature extraction technique with an RMSE value of 3.4 %, for the synthetic dataset. Transfer learning results were not more reliable than those achieved through feature extraction using traditional feature extraction or pretrained ConvNet.

According to a real dataset, all bands and sizes of images have shown impressive results using only the top view of soils. Specifically, PSDNet could predict PSD with an RMSE of 3.7% for color images and 3.9% for grayscale images. The best results were obtained from the features extracted from InceptionResNetV2, and a combination of PSDNET Gray and Color extracted features with RMSE values of 1.7 and 1.8%, respectively. Furthermore, the combination of all color PSDNet (500 components) and traditional features (1208 components) could result in an RMSE value of 2.3%. According to the material removal test results, average RMSE values varied between 2.8 and 19.2%. These techniques prove their effectiveness in predicting unseen material in real projects. Consequently, PSDNet is capable of predicting real soil based on the top view.

Training is a key difference between neural networks and image segmentation methods. Models based on image segmentation only analyze the particles' surface. In addition, neural network-based techniques, such as ConvNets and ANN, can be improved through training during their use. Model accuracy can also be improved by extending the dataset for neural network-based models.

The purpose of this study was to examine several methods of predicting the soil PSD, including pretrained ConvNet feature extraction, traditional feature extraction, pretrained ConvNet using transfer learning, transfer learning-based feature extraction, and a combination of all feature extraction techniques that can accurately predict the soil PSD. The combination of feature extraction techniques led to better results.

6.2 RECOMMENDATIONS

Other applications can also be considered; for example, similar techniques can be utilized for soil classification (e.g., ASTM D2487) and the estimation of soil properties (e.g., hydraulic conductivity, compaction, compressibility, and shear strength).

The Unity game engine and GANs (Generative Adversarial Networks) (Frid-Adar et al., 2018) can provide more realistic synthetic particle images for fine-tuning the model for a new real soil with low labelled images. In this case, it is good to use the 3D scan of target soil as a skin of simulated soil particles. Then, a combination of real and simulated images could be used to train the model. A more extensive database helps improve the quality of the training process.

It is recommended to examine techniques introduced in this thesis for a broader spectrum of real datasets in future works. For example, it can be used for very coarse materials. The Aerial images can be taken by UAV or satellite to capture inaccessible rocks. After labelling and training the model, it can be easy to predict the size of these particles. Using a combination of different image technics introduced in this thesis can increase the accuracy of predicting particle size. Because labelling of big rocks is very difficult, using synthetic images can also be helpful in this case. Usually, analyzing this type of particle is too complicated and expensive manually (Bouchard, 2016).

Other capturing systems can be utilized to train ConvNet and predict PSD. For instance, depth camera RGB-D (Nguyen, Kim, Kim, and Na, 2017) and point cloud like LiDAR (Kashefi and Mukerji, 2021; Shi, Zhong, and Bi, 2021; Xing, Zhao, Wang, Nie, and Du, 2021) are the two recommended technics that can improve the results of particle sizing.

There are numerous methods for image feature extraction (Liu et al., 2019). In this thesis, nine methods of traditional methods were validated to find PSD. Other types of feature extraction techniques can examine to predict PSD like Filters by Leung and Malik (LMFilters) (Leung and Malik, 2001), Schmid Filters (S Filters) (Schmid, 2001), Maximum Response (MR8) Filters (Varma and Zisserman, 2005), Patch Descriptors (Varma and Zisserman, 2009), Random Projection (RP) and Sorted Random Projection (SRP) (Li Liu and Fieguth, 2012), Basic Image Features (BIF) (Crosier and Griffin, 2010), Weber Law Descriptor (WLD) (Jie Chen et al., 2010).

In addition, it is recommended to examine some object detection methods like Speeded-Up Robust Features (SURF)(Bay, Ess, Tuytelaars, and Van Gool, 2008) and scale-invariant feature transform (SIFT)(Lowe, 1999). SURF is a type of local feature extractor return point object with descriptors. This method is patented approximated Laplacian of Gaussian with Box filter, which could faster be calculated. SUFT is relatively similar to SIFT but three times faster (Banerjee and Mistry, 2017). SURF and SIFT are invariant in transformation, rotating and blurring of the image.

There are other convolutional architectures and convolutional block structures that can also be considered for improving the PSDNet model. Using residual blocks (He et al., 2016) as a block structure can improve the PSDNet model. Meanwhile, PSDNet is focused primarily on deploying the depth of network. It has also been suggested that the width of the network is important by Kawaguchi et al. (2019). ConvNet can be interconnected using multi-path or cross-layer connections. In order to provide a continuous flow of information across the layers, one can systematically connect the layers through multiple paths or shortcuts by skipping some intermediate layers (Wang, 2015)

.

LIST OF BIBLIOGRAPHICAL REFERENCES

AASHTO. 2021. M 145—Standard Specification for Classification of Soils and Soil–Aggregate Mixtures for Highway Construction Purposes. In *American Association of State Highway and Transportation Officials (AASHTO)*.

Abbireddy, C. O. R., & Clayton, C. R. I. 2009. A review of modern particle sizing methods. *Proceedings of the Institution of Civil Engineers - Geotechnical Engineering*, *162*(4), 193–201. https://doi.org/10.1680/geng.2009.162.4.193

Abdel-Hamid, O., Deng, L., & Yu, D. 2013. Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition. *14th Annual Conference of the International Speech Communication Association, (Interspeech 2013)*, *11*(August), 73–75.

Abdel-Hamid, O., Mohamed, A., Jiang, H., & Penn, G. 2012. Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4277–4280. https://doi.org/10.1109/ICASSP.2012.6288864

Al-Thyabat, S., Miles, N. J., & Koh, T. S. 2007. Estimation of the size distribution of particles moving on a conveyor belt. *Minerals Engineering*, *20*(1), 72–83. https://doi.org/10.1016/j.mineng.2006.05.011

Allen, T. 2003. *Powder sampling and particle size determination*. Elsevier, Amsterdam.

ASTM International. 2007. D422 Standard Test Method for Particle-Size Analysis of Soils. In *West Conshohocken, PA* (Issue Reapproved). https://doi.org/10.1520/D0422-63R07E01

ASTM International. 2017. D2487-17e1 Standard Practice for Classification of Soils for Engineering Purposes (Unified Soil Classification System). In *West Conshohocken, PA*. https://doi.org/10.1520/D2487-17E01

ASTM International. 2019a. C136/C136M-19 Standard Test Method for Sieve Analysis of Fine and Coarse Aggregates. In *West Conshohocken, PA*. https://doi.org/10.1520/C0136_C0136M-19

ASTM International. 2019b. D6913/D6913M-17 Standard Test Methods for Particle-Size Distribution (Gradation) of Soils Using Sieve Analysis. In *West Conshohocken, PA*. https://doi.org/10.1520/D6913_D6913M-17

ASTM International. 2021. D7928-21e1 Standard Test Method for Particle-Size Distribution (Gradation) of Fine-Grained Soils Using the Sedimentation (Hydrometer) Analysis. In *West Conshohocken, PA*. https://doi.org/10.1520/D7928-21E01

Banerjee, A., & Mistry, D. 2017. Comparison of Feature Detection and Matching Approaches: SIFT and SURF. *GRD Journals-Global Research and Development Journal for Engineering*, *2*(March), 7–13. https://www.researchgate.net/publication/314285930

Bardet, J.-P. 1997. *Experimental soil mechanics*. Prentice Hall Upper Saddle River, NJ.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359. https://doi.org/10.1016/j.cviu.2007.09.014

Blanc-Durand, P. 2018. *Réseaux de neurones convolutifs en médecine nucléaire : Applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM. (Doctoral thesis, Université de Paris, Paris, France).*

https://dumas.ccsd.cnrs.fr/dumas-02116080/document

Bouchard, J. 2016. *Caractérisation de l ' enrochement du barrage Romaine -2. (Master's thesis, Université de Laval, Quebec City, QC)*. https://corpus.ulaval.ca/jspui/handle/20.500.11794/17461

British Standards Institution. 1990. *British Standard Methods of Test for Soils for Civil Engineering Purposes*. British Standards Institution.

Buscombe, D. 2020. SediNet: a configurable deep learning model for mixed qualitative and quantitative optical granulometry. *Earth Surface Processes and Landf.*, *45*(3), 638–651. https://doi.org/10.1002/esp.4760

Caputo, F., Clogston, J., Calzolai, L., Rösslein, M., & Prina-Mello, A. 2019. Measuring particle size distribution of nanoparticle enabled medicinal products, the joint view of EUNCL and NCI-NCL. A step by step approach combining orthogonal measurements with increasing complexity. *Journal of Controlled Release*, *299*(February), 31–43. https://doi.org/10.1016/j.jconrel.2019.02.030

Carrier, W. D. 2003. Particle Size Distribution of Lunar Soil. *Journal of Geotechnical and Geoenvironmental Engineering*, *129*(10), 956–959. https://doi.org/10.1061/(ASCE)1090-0241(2003)129:10(956)

Center for Machine Vision Research, Department of Computer Science and Engineering, Matlab code. n.d. University in Oulu, Finland. https://www.oulu.fi/cmvs/node/33019

Chakraborty, M., Biswas, S. K., & Purkayastha, B. 2019. Rule Extraction from Neural Network Using Input Data Ranges Recursively. *New Generation Computing*, *37*(1), 67–96. https://doi.org/10.1007/s00354-018-0048-0

Chavez, R., Cheimanoff, N., & J. Schleifer. 1996. Sampling problems during grain size distribution measurements. *Proceedings of the Fifth Inter- National Symposium on Rock Fragmentation by Blasting – FRAGBLAST*, 245–252.

Cheimanoff, N. M., Chavez, R., & Schleifer, J. 1993. Fragscan: A scanning tool for fragmentation after blasting. *International Symposium on Rock Fragmentation by Blasting*, 325–330.

Chellapilla, K., Puri, S., & Simard, P. 2006. High Performance Convolutional Neural Networks for Document Processing. *Tenth International Workshop on Frontiers in Handwriting Recognition*, 1–6. https://hal.inria.fr/inria-00112631

Chollet, F. 2016. Xception: Deep Learning with Depthwise Separable Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251–1258. http://arxiv.org/abs/1610.02357

Chollet, F. 2017. Deep Learning with Python. In *Manning Publications Co.* Apress.

Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. 2012. Multi-column deep neural network for traffic sign classification. *Neural Networks*, *32*, 333–338. https://doi.org/10.1016/j.neunet.2012.02.023

Costodes, V. C. T., Mausse, C. F., Molala, K., & Lewis, A. E. 2006. A simple approach for determining particle size enlargement mechanisms in nickel reduction. *International Journal of Mineral Processing*, *78*(2), 93–100. https://doi.org/10.1016/j.minpro.2005.09.001

Crosier, M., & Griffin, L. D. 2010. Using Basic Image Features for Texture Classification. *International Journal of Computer Vision*, *88*(3), 447–460. https://doi.org/10.1007/s11263-009-0315-0

Cundall, P. A. ., & Strack, O. D. L. 1979. A discrete numerical model for granular assemblies. *Géotechnique*, *29*(1), 47–65. https://doi.org/10.1680/geot.1979.29.1.47

Dahl, G. E., Sainath, T. N., & Hinton, G. E. 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8609–8613. https://doi.org/10.1109/ICASSP.2013.6639346

Dalal, N., & Triggs, B. 2005. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, *May 2014*, 248–255. https://doi.org/10.1109/CVPRW.2009.5206848

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

Detert, M., & Weitbrecht, V. 2012. Automatic object detection to analyze the geometry of gravel grains--a free stand-alone tool. *River Flow*, 595–600.

Dubé, J.-S., Ternisien, J., Boudreault, J.-P., Duhaime, F., & Éthier, Y. 2021. Variability in Particle Size Distribution Due to Sampling. *Geotechnical Testing Journal*, *44*(1), 20190030. https://doi.org/10.1520/GTJ20190030

Duhaime, F., Pirnia, P., Manashti, J., Temimi, M., & Toews, M. 2021. Particle size distribution from photographs : comparison of synthetic and real granular material images. *GeoNiagara*.

Dumberry, K., Duhaime, F., & Ethier, Y. A. 2018. Erosion monitoring during core overtopping using a laboratory model with digital image correlation and X-ray microcomputed tomography. *Canadian Geotechnical Journal*, *55*(2), 234–245. https://doi.org/10.1139/cgj-2016-0684

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. 2018. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, *321*, 321–331. https://doi.org/https://doi.org/10.1016/j.neucom.2018.09.013

Fu, Y., & Aldrich, C. 2018. Froth image analysis by use of transfer learning and convolutional neural networks. *Minerals Engineering*, *115*(July 2017), 68–78. https://doi.org/10.1016/j.mineng.2017.10.005

Gadkari, D. 2004. *Image Quality Analysis Using GLCM. (Master's thesis, University of Central Florida, Orlando, United States)*. http://purl.fcla.edu/fcla/etd/CFE0000273

Gatys, L. A., Ecker, A. S., & Bethge, M. 2016. Image Style Transfer Using Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423. https://doi.org/10.1109/CVPR.2016.265

Gee, G. W., & Bauder, J. W. 1986. Particle-size analysis. In *Methods of soil analysis: Part 1—Physical*. pp. 383–411. Soil Science Society of America, American Society of Agronomy.

Ghalib, A. M., Hryciw, R. D., & Shin, S. C. 1998. Image Texture Analysis and Neural Networks for Characterization of Uniform Soils. *Computing in Civil Engineering*, 671–682.

Gholamy, A., Kreinovich, V., & Kosheleva, O. 2018. Why 70/30 or 80/20 Relation Between Training and Testing Sets : A Pedagogical Explanation. *Departmental Technical Reports (CS)*, 1–6.

Girdner, K. K., Kemeny, J. M., Srikant, A., & McGill, R. 1996. The split system for analyzing the size distribution of fragmented rock. *Proceedings , FRAGBLAST-5 Workshop on Measurement of Blast Fragmentation, Edited by J. A. Franklin and T. Katsabanis, 101-108. Rotterdam, NL: Balkema*, 101–108.

Gonzalez, R. C., & Woods, R. E. 2002. *Digital Image Processing, 2nd Edition.* Pearson India.

Gonzalez, R. C., & Woods, R. E. 2018. *Digital image processing, 4th Edition*. Pearson India.

Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. MIT press.

Guo, Y., Zhao, G., & Pietikäinen, M. 2011. Texture Classification using a Linear Configuration Model based Descriptor. *Procedings of the British Machine Vision Conference 2011*, 119.1-119.10. https://doi.org/10.5244/C.25.119

Guo, Y., Zhao, G., & Pietikäinen, M. 2012. Discriminative features for texture description. *Pattern Recognition*, *45*(10), 3834–3843. https://doi.org/10.1016/j.patcog.2012.04.003

Guo, Z., Zhang, L., & Zhang, D. 2010. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Transactions on Image Processing*, *19*(6), 1657–1663. https://doi.org/10.1109/TIP.2010.2044957

Guyot, O., Monredon, T., LaRosa, D., & Broussaud, A. 2004. VisioRock, an integrated vision technology for advanced control of comminution circuits. *Minerals Engineering*, *17*(11–12), 1227–1235. https://doi.org/10.1016/j.mineng.2004.05.017

Haas, J. 2014. A History of the Unity Game Engine: An Interactive Qualifying Project. In *Worcester Polytechnic Institute*.

Hall, S. A., Bornert, M., Desrues, J., Pannier, Y., Lenoir, N., Viggiani, G., & Bésuelle, P. 2010. Discrete and continuum analysis of localised deformation in sand using X-ray μCT and volumetric digital image correlation. *Géotechnique*, *60*(5), 315–322. https://doi.org/10.1680/geot.2010.60.5.315

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

Haralick, R. M., Shanmugam, K., & Dinstein, I. 1973. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

Haustein, M., Gladkyy, A., & Schwarze, R. 2017. Discrete element modeling of deformable particles in YADE. *SoftwareX*, *6*, 118–123. https://doi.org/10.1016/j.softx.2017.05.001

He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://arxiv.org/abs/1512.03385

Herman, B. M., Browning, S. R., & Reagan, J. A. 1971. Determination of Aerosol Size Distributions from Lidar Measurements. *Journal of the Atmospheric Sciences*, *28*(5), 763–771. https://doi.org/10.1175/1520-0469(1971)028<0763:DOASDF>2.0.CO;2

Hryciw, R. D., Ohm, H.-S., & Zhou, J. 2015. Theoretical Basis for Optical Granulometry by Wavelet Transformation. *Journal of Computing in Civil Engineering*, *29*(3), 04014050. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000345

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. 2017. Densely Connected

Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *ArXiv Preprint ArXiv:1602.07360*. http://arxiv.org/abs/1602.07360

Ilonen, J., Juránek, R., Eerola, T., Lensu, L., Dubská, M., Zemčík, P., & Kälviäinen, H. 2018. Comparison of bubble detectors and size distribution estimators. *Pattern Recognition Letters*, *101*, 60–66. https://doi.org/10.1016/j.patrec.2017.11.014

Ioffe, S., & Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning*, 448–456.

Itoh, H., Matsuo, K., Oida, A., Nakashima, H., Miyasaka, J., & Izumi, T. 2008. Aggregate size measurement by machine vision. *Journal of Terramechanics*, *45*(4), 137–145. https://doi.org/10.1016/j.jterra.2008.09.001

Jackson, G. A., Logan, B. E., Alldredge, A. L., & Dam, H. G. 1995. Combining particle size spectra from a mesocosm experiment measured using photographic and aperture impedance (Coulter and Elzone) techniques. *Deep Sea Research Part II: Topical Studies in Oceanography*, *42*(1), 139–157. https://doi.org/10.1016/0967-0645(95)00009-F

Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Pietikäinen, M., Xilin Chen, & Wen Gao. 2010. WLD: A Robust Local Image Descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1705–1720. https://doi.org/10.1109/TPAMI.2009.155

Jung, Y. 2010. *Determination of Soil Grain Size Distribution by Soil Sedimentation and Image Processing. (Doctoral thesis, University of Michigan, Ann Arbor, United States)*. https://hdl.handle.net/2027.42/77895

Karpathy, A. n.d. *CS231n: Convolutional Neural Networks for Visual Recognition*. http://cs231n.github.io/convolutional-networks/#layers

Kashefi, A., & Mukerji, T. 2021. *Point-Cloud Deep Learning of Porous Media for Permeability Prediction*. http://arxiv.org/abs/2107.14038

Kavukcuoglu, K., Sermanet, P., Boureau, Y., Gregor, K., Mathieu, M., Cun, Y. L., & LeCun, Y. 2010. Learning Convolutional Feature Hierarchies for Visual Recognition. *Advances in Neural Information Processing Systems 23*, *1*, 1090–1098.

Kawaguchi, K., Huang, J., & Kaelbling, L. P. 2019. Effect of Depth and Width on Local Minima in Deep Learning. *Neural Computation*, *31*(7), 1462–1498. https://doi.org/10.1162/neco_a_01195

Kaya, Y., Ertuğrul, Ö. F., & Tekin, R. 2015. Two novel local binary pattern descriptors for texture analysis. *Applied Soft Computing*, *34*, 728–735. https://doi.org/10.1016/j.asoc.2015.06.009

Khellaf, A., Beghdadi, A., & Dupoisot, H. 1991. Entropic contrast enhancement. *IEEE Transactions on Medical Imaging*, *10*(4), 589–592. https://doi.org/10.1109/42.108593

Ko, Y.-D., & Shang, H. 2011a. Time delay neural network modeling for particle size in SAG mills. *Powder Technology*, *205*(1–3), 250–262. https://doi.org/10.1016/j.powtec.2010.09.023

Ko, Y.-D., & Shang, H. 2011b. A neural network-based soft sensor for particle size distribution using image analysis. *Powder Technology*, *212*(2), 359–366. https://doi.org/10.1016/j.powtec.2011.06.013

Kozicki, J., & Donzé, F. V. 2009. YADE-OPEN DEM: an open-source software using a discrete element method to simulate granular material. *Engineering Computations*, *26*(7), 786–805. https://doi.org/10.1108/02644400910985170

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Lang, N., Irniger, A., Rozniak, A., Hunziker, R., Wegner, J. D., & Schindler, K. 2021. GRAINet: mapping grain size distributions in river beds from UAV images with convolutional neural networks. *Hydrology and Earth System Sciences*, *25*(5), 2567–2597. https://doi.org/10.5194/hess-25-2567-2021

Larkin, K. G. 2016. Reflections on Shannon Information: In search of a natural information-entropy for images. *In Search of a Natural Information-Entropy for Images*. http://arxiv.org/abs/1609.01117

LeCun, Y., & Bengio, Y. 1995. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10).

Lee, J.-G., Jun, S., Cho, Y., Lee, H., Kim, G. B., Seo, J. B., & Kim, N. 2017. Deep Learning in Medical Imaging: General Overview. *Korean Journal of Radiology*, *18*(4), 570. https://doi.org/10.3348/kjr.2017.18.4.570

Lee, W.-J., Kim, D., Kang, T.-K., & Lim, M.-T. 2018. Convolution Neural Network with Selective Multi-Stage Feature Fusion: Case Study on Vehicle Rear Detection. *Applied Sciences*, *8*(12), 2468. https://doi.org/10.3390/app8122468

Leung, T., & Malik, J. 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, *43*(1), 29–44. https://doi.org/https://doi.org/10.1023/A:1011126920638

Li Liu, & Fieguth, P. 2012. Texture Classification from Random Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(3), 574–586. https://doi.org/10.1109/TPAMI.2011.145

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. *International Journal of Computer Vision*, *127*(1), 74–109. https://doi.org/10.1007/s11263-018-1125-z

Liu, L., Ji, M., & Buchroithner, M. 2018. Transfer learning for soil spectroscopy based on convolutional neural networks and its application in soil clay content mapping using hyperspectral imagery. *Sensors (Switzerland)*, *18*(9). https://doi.org/10.3390/s18093169

Liu, Q., & Tran, H. 1996. Comparing systems - Validation of FragScan, WipFrag and Split. *Proceedings of the Fifth International Symposium on Rock Fragmentation by Blasting – FRAGBLAST*, 151–155.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1150–1157 vol.2. https://doi.org/10.1109/ICCV.1999.790410

Maerz, N. H., Palangio, T. C., & Franklin, J. A. 1996. WipFrag image based granulometry system. *Proceedings of the Fifth International Symposium on Rock Fragmentation by Blasting – FRAGBLAST*, *5*, 91–99.

Maerz, N. H., T.C. Palangio, T.W. Palangio, & K. Elsey. 2007. Optical sizing analysis of blasted rock: lessons learned. *Materiały Konferencyjne 4th EFEE World Conference of Explosives and Blasting in Vienna*, 75–83.

Manashti, J., Duhaime, F., Toews, M. F., & Pirnia, P. 2021. Predicting the Particle Size Distributions of Spherical Particle Sets from Synthetic Images : A Comparison of 9 Classic Image Features. *EngrXiv*, *2*. https://doi.org/https://doi.org/10.31224/osf.io/phjkm

Manashti, J., Pirnia, P., Manashty, A., Ujan, S., Toews, M., & Duhaime, F. 2022. PSDNet: Determination of Particle Size Distributions Using Synthetic Soil Images and Convolutional Neural Networks. s*ubmitted to Acta Geotechnica*.

Manashti, M. J. 2022. *Grain Size Analyses of Soils Based on Image Analysis Techniques and Machine Learning. (Doctoral Dissertation, École de Technologie Supérieure, Montreal, QC).*

Manjunathi, B. S., & Ma, W. Y. 1996. Texture features for browsing and retrieval of image data. *IEEE T. Pattern Anal.*, *18*(8), 837–842.

Maren, A. J., Harston, C. T., & Pap, R. M. 2014. *Handbook of neural computing applications*. Academic Press.

MATLAB. 2019. *2019*. "ExtractHOGFeatures."; The MathWorks Inc. https://www.mathworks.com/help/vision/ref/extracthogfeatures.html

McFall, B., Young, D., Fall, K., Krafft, D., Whitmeyer, S., Melendez, A., & Buscombe, D. 2020. Technical feasibility of creating a beach grain size database with citizen scientists. *Coastal and Hydraulics Laboratory (U.S.), Engineer Research and Development Center (U.S.)*. https://doi.org/10.21079/11681/36456

Monzel, R. 2019. *haralickTextureFeatures*. MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/58769-haralicktexturefeatures

Ng, W., Minasny, B., & McBratney, A. 2020. Convolutional neural network for soil microplastic contamination screening using infrared spectroscopy. *Science of the Total Environment*, *702*, 134723. https://doi.org/10.1016/j.scitotenv.2019.134723

Nguyen, H. D., Kim, Y. C., Kim, S. H., & Na, I. S. 2017. A method for fingertips detection using RGB-D image and convolution neural network. *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 783–785. https://doi.org/10.1109/FSKD.2017.8393373

Nikolenko, S. I. 2021. Synthetic data for deep learning. In *Springer*.

Ohm, H.-S. n.d. *Image-Based Soil Particle Size and Shape Characterization. (PhD diss., University of Michigan, 2013)*.

Ohm, H.-S., & Hryciw, R. D. 2014a. Soil Fabric Characterization by Wavelet Transformation of Images. *Geo-Congress 2014 Technical Papers*, 723–730. https://doi.org/10.1061/9780784413272.070

Ohm, H.-S., & Hryciw, R. D. 2014b. Size Distribution of Coarse-Grained Soil by Sedimaging. *Journal of Geotechnical and Geoenvironmental Engineering*, *140*(4), 04013053. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001075

Ojala, T., Pietikainen, M., & Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. 2014. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1717–1724. https://doi.org/10.1109/CVPR.2014.222

Paclík, P., Verzakov, S., & Duin, R. P. W. 2005. Improving the Maximum-Likelihood Co-occurrence Classifier: A Study on Classification of Inhomogeneous Rock Images. In *Lecture Notes in Computer Science* (Vol. 3540, pp. 998–1008). https://doi.org/10.1007/11499145_101

Padarian, J., Minasny, B., & McBratney, A. B. 2019. Transfer learning to localise a continental soil vis-NIR calibration model. *Geoderma*, *340*, 279–288. https://doi.org/10.1016/j.geoderma.2019.01.009

Pang, Y., Yuan, Y., Li, X., & Pan, J. 2011. Efficient HOG human detection. *Signal Processing*, *91*(4), 773–781. https://doi.org/10.1016/j.sigpro.2010.08.010

Park, S., & Kwak, N. 2017. Analysis on the Dropout Effect in Convolutional Neural Networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10112 LNCS* (pp. 189–204). https://doi.org/10.1007/978-3-319-54184-6_12

Patel, A. K., Chatterjee, S., & Gorai, A. K. 2019. Development of a machine vision system using the support vector machine regression (SVR) algorithm for the online prediction of iron ore grades. *Earth Science Informatics*, *12*(2), 197–210. https://doi.org/10.1007/s12145-018-0370-6

Peng, Y., Cai, J., Wu, T., Cao, G., Kwok, N., Zhou, S., & Peng, Z. 2019. A hybrid convolutional neural network for intelligent wear particle classification. *Tribology International*, *138*, 166–173. https://doi.org/10.1016/j.triboint.2019.05.029

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. *Computers & Geosciences*, *123*, 38–46. https://doi.org/10.1016/j.cageo.2018.11.002

Pirnia, P., Duhaime, F., & Manashti, J. 2018. Machine learning algorithms for applications in geotechnical engineering. *71st Canadian Geotechnical Conference*, 7 pp. (paper 339).

Plautz, T., Boudreau, R., Chen, J.-H., Ekman, A., LeGros, M., McDermott, G., & Larabell, C. 2017. Progress Toward Automatic Segmentation of Soft X-ray Tomograms Using Convolutional Neural Networks. *Microscopy and Microanalysis*, *23*(S1), 984–985. https://doi.org/10.1017/S143192761700558X

Potts, G., & Ouchterlony, F. 2005. *The capacity of image analysis to measure fragmentation: An evaluation using Split Desktop*. Technical report, Swedish Blasting Research Centre och Luleå tekniska universitet.

Prats-Montalbán, J. M., de Juan, A., & Ferrer, A. 2011. Multivariate image analysis: A review with applications. *Chemometrics and Intelligent Laboratory Systems*, *107*(1), 1–23. https://doi.org/10.1016/j.chemolab.2011.03.002

Raina, A. K., Choudhury, P. B., Ramulu, M., Chakraborty, A. K., & Misra, D. D. 2002. Fragalyst-An indigenous digital image analysis system for grain size measurements in mines. *Journal of the Geological Society of India*, *59*(6), 561–569.

Rajpura, P. S., Bojinov, H., & Hegde, R. S. 2017. Object Detection Using Deep CNNs Trained on Synthetic Images. *ArXiv Preprint ArXiv:1706.06782*. http://arxiv.org/abs/1706.06782

Raza Ansari, S. 2018. *Application of Machine Learning Techniques for Soil Type Classfication of Karanataka. National College of Ireland. (Master's thesis, National College of Ireland, Dublin, Ireland).* (Issue July). http://trap.ncirl.ie/id/eprint/3443

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. 2015. ImageNet Large Scale Visual

Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Saghatoleslam, N., Karimi, H., Rahimi, R., & Shirazi, H. H. A. 2004. Modeling of texture and color froth characteristics for evaluation of flotation performance in sarcheshmeh copper pilot plant using image analysis and neural networks. *International Journal of Engineering, Transactions B: Applications*, *17*(2), 121–130.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Scherer, D., Müller, A., & Behnke, S. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6354 LNCS*(PART 3), 92–101. https://doi.org/10.1007/978-3-642-15825-4_10

Schmid, C. 2001. Constructing models for content-based image retrieval. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *2*, II-39-II–45. https://doi.org/10.1109/CVPR.2001.990922

Schneider, C. L., Neumann, R., & Souza, A. S. 2007. Determination of the distribution of size of irregularly shaped particles from laser diffractometer measurements. *International Journal of Mineral Processing*, *82*(1), 30–40. https://doi.org/10.1016/j.minpro.2006.09.011

Sereshki, F., Hoseini, M., & Ataei, M. 2016. Blast fragmentation analysis using image processing. *International Journal of Mining and Geo-Engineering*, *50*(2), 211–218. https://doi.org/10.22059/ijmge.2016.59831

Sermanet, P., & LeCun, Y. 2011. Traffic sign recognition with multi-scale Convolutional Networks. *The 2011 International Joint Conference on Neural Networks*, *September 2011*, 2809–2813. https://doi.org/10.1109/IJCNN.2011.6033589

Shi, T., Zhong, D., & Bi, L. 2021. A New Challenge: Detection of Small-Scale Falling Rocks on Transportation Roads in Open-Pit Mines. *Sensors*, *21*(10), 3548. https://doi.org/10.3390/s21103548

Shin, S., & Hryciw, R. D. 2004. Wavelet Analysis of Soil Mass Images for Particle Size Determination. *Journal of Computing in Civil Engineering*, *18*(1), 19–27. https://doi.org/10.1061/(ASCE)0887-3801(2004)18:1(19)

Simonyan, K., & Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations (ICLR)*, 1–14. http://arxiv.org/abs/1409.1556

Šmilauer, V., Catalano, E., Chareyre, B., Sergei, Eulitz, A., Gladky, A., Guo, N., Jakob, C., Dorofeenko, F., Duriez, J., Dyck, N., Eliáš, J., Er, B. M., Scholtès, L., Sibille, L., Stránský, J., Kneib, T. S., Kozicki, J., Marzougui, D., … Yuan, C. 2015. Yade Documentation 2nd ed. In *The Yade Project*. https://doi.org/10.5281/zenodo.34073

Split Engineering. 2021. *Split Desktop*. https://www.spliteng.com/products/split-desktop-software/

Srivastava, N., Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, & Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Stokes, G. G. 1891. *Mathematical and physics papers, 3, 55*. Cambridge University Press.

Sudha, D., & Ramakrishna, M. 2017. Comparative Study of Features Fusion Techniques. *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, 235–239. https://doi.org/10.1109/ICRAECT.2017.39

Sudhakar, J., Adhikari, G. R., & Gupta, R. N. 2006. Comparison of Fragmentation Measurements by Photographic and Image Analysis Techniques. *Rock Mechanics and Rock Engineering*, *39*(2), 159–168. https://doi.org/10.1007/s00603-005-0044-9

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Thirty-First AAAI Conference on Artificial Intelligence*, 4278–4284. http://arxiv.org/abs/1602.07261

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *07-12-June*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Szeliski, R. 2011. Computer Vision, Algorithms and Applications. In *Springer Tracts in Advanced Robotics* (Vol. 73). Springer. https://doi.org/10.1007/978-1-84882-935-0

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. 2016. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, *35*(5), 1299–1312. https://doi.org/10.1109/TMI.2016.2535302

Telcy, J. K. 2018. *Granulométries des sols à partir des caractéristiques texturales d'Haralick. (Master's Project Report, École de technologie supérieure, Montréal, QC)*. https://espace.etsmtl.ca/view/

Temimi, M. 2020. *Création d ' images de matériaux granulaires avec la plateforme de développement de jeux Unity et prédiction de la granulométrie à l ' aide de textures et de réseaux de neurones. (Master's thesis, École de technologie supérieure, Montréal, QC)*. https://espace.etsmtl.ca/view/

Thurley, M. J. 2011. Automated online measurement of limestone particle size distributions using 3D range data. *Journal of Process Control*, *21*(2), 254–262. https://doi.org/10.1016/j.jprocont.2010.11.011

Thurley, M. J., & Ng, K. C. 2008. Identification and sizing of the entirely visible rocks from a 3D surface data segmentation of laboratory rock piles. *Computer Vision and Image Understanding*, *111*(2), 170–178. https://doi.org/10.1016/j.cviu.2007.09.009

Tian, D. P. 2013. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, *8*(4), 385–395.

Tuceryan, M., & Jain, A. K. 1993. Texture Analysis. In *Handbook of Pattern Recognition and Computer Vision* (pp. 235–276). WORLD SCIENTIFIC. https://doi.org/10.1142/9789814343138_0010

Unity Technologies. 2017. *Unity User Manual (2017.3)*. https://docs.unity3d.com/Manual/index.html

University of Queensland. 2016. *Introduction to Unity*. https://www.eait.uq.edu.au/filething/get/20519/INTRODUCTION TO UNITY.pdf

Varma, M., & Zisserman, A. 2005. A Statistical Approach to Texture Classification from Single Images. *International Journal of Computer Vision*, *62*(1/2), 61–81. https://doi.org/10.1023/B:VISI.0000046589.39864.ee

Varma, M., & Zisserman, A. 2009. A Statistical Approach to Material Classification Using Image Patch Exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(11), 2032–2047. https://doi.org/10.1109/TPAMI.2008.182

Vinet, L., & Zhedanov, A. 2011. A 'missing' family of classical orthogonal polynomials. Journal of Physics A: Mathematical and Theoretical, 44(8), 085201. https://doi.org/10.1088/1751-8113/44/8/085201

Wang, M. 2015. *Multi-path Convolutional Neural Networks for Complex Image Classification*. http://arxiv.org/abs/1506.04701

Weiss, K., Khoshgoftaar, T. M., & Wang, D. 2016. A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. https://doi.org/10.1186/s40537-016-0043-6

Wen, B., Aydin, A., & Duzgoren-Aydin, N. S. 2002. A comparative study of particle size analyses by sieve-hydrometer and laser diffraction methods. *Geotechnical Testing Journal*, *25*(4), 434–442.

White, D. J. 2003. PSD measurement using the single particle optical sizing (SPOS) method. *Géotechnique*, *53*(3), 317–326. https://doi.org/10.1680/geot.2003.53.3.317

Wu, S., Zhong, S., & Liu, Y. 2018. Deep residual learning for image steganalysis. *Multimedia Tools and Applications*, *77*(9), 10437–10453. https://doi.org/10.1007/s11042-017-4440-4

Xing, J., Zhao, Z., Wang, Y., Nie, L., & Du, X. 2021. Coal and gangue identification method based on the intensity image of lidar and DenseNet. *Applied Optics*, *60*(22), 6566. https://doi.org/10.1364/AO.422498

Yaghoobi, H., Mansouri, H., Ebrahimi Farsangi, M. A., & Nezamabadi-Pour, H. 2019. Determining the fragmented rock size distribution using textural feature extraction of images. *Powder Technology*, *342*, 630–641. https://doi.org/10.1016/j.powtec.2018.10.006

Yakura, H., Shinozaki, S., Nishimura, R., Oyama, Y., & Sakuma, J. 2018. Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism. *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, *2018-Janua*, 127–134. https://doi.org/10.1145/3176258.3176335

Yan, K., Lu, L., & Summers, R. M. 2018. Unsupervised body part regression via spatially self-ordering convolutional neural networks. *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 1022–1025. https://doi.org/10.1109/ISBI.2018.8363745

Yang, J., & Chen, S. 2017. An online detection system for aggregate sizes and shapes based on digital image processing. *Mineralogy and Petrology*, *111*(1), 135–144. https://doi.org/10.1007/s00710-016-0458-y

Yu, D., Eversole, A., Seltzer, M., Yao, K., Huang, Z., Guenter, B., Kuchaiev, O., Zhang, Y., Seide, F., Wang, H., Droppo, J., Zweig, G., Rossbach, C., Currey, J., Gao, J., May, A., Peng, B., Stolcke, A., & Slaney, M. 2014. An Introduction to Computational Networks and the Computational Network Toolkit (Invited Talk). *15th Annual Conference of the International Speech Communication Association*.

Zeiler, M. D., & Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 8689 LNCS (Issue PART 1, pp. 818–833). https://doi.org/10.1007/978-3-319-10590-1_53

Zhang, X., Zhou, X., Lin, M., & Sun, J. 2017. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6848–6856. http://arxiv.org/abs/1707.01083

Zhu, Y., Ouyang, Q., & Mao, Y. 2017. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. *BMC Bioinformatics*, *18*(1), 348. https://doi.org/10.1186/s12859-017-1757-y

Zobeck, T. M. 2004. Rapid Soil Particle Size Analyses Using Laser Diffraction. *Applied Engineering in Agriculture*, *20*(5), 633–639. https://doi.org/10.13031/2013.13799

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. 2018. Learning transferable architectures for scalable image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8697–8710.