Deep Learning Assisted Particle Identification in Photoelastic Images of Granular Flows

Nicoletta Sanvitale[1*], Claudio Gheller[2] and Elisabeth Bowman[3]

[1*]Department of Civil, Chemical, Environmental, and Materials Engineering, University of Bologna, Viale Risorgimento 2, Bologna, 40136, Italy.
[2]Institute of Radioastronomy, INAF, Via Gobetti 101, Bologna, 40127, Italy.
[3]Department of Civil and Structural Engineering, University of Sheffield, Mappin St, Sheffield, S1 3JD, United Kingdom.

*Corresponding author(s). E-mail(s): nicoletta.sanvitale@gmail.com;

Abstract

The transmission of forces within high speed granular flows may be straightforwardly viewed in two-dimensional photoelastic experiments, but precise measurements have remained elusive due to difficulties in differentiating between particles and forces with sufficient accuracy at reasonable processing speeds. This paper presents a novel approach to detect the positions of disks embedded in this complex situation, which is a crucial step in applying the methodologies necessary for the analysis of the photoelastic response of individual disks. We have developed a Deep Learning based solution to perform the segmentation of experimental photoelastic images, disentangling with high fidelity the disk outlines from the rest of each image. The accuracy and the reliability of the proposed methodology are discussed in detail, demonstrating that this approach can be effectively adopted for the problem under investigation, improving the quality of the photoelastic analysis and dramatically accelerating the data processing procedure.

Keywords: Granular Flows, Photoelasticity, Machine Learning, Image analysis, Particle tracking

1 Introduction

The study of granular flow can adopt different configurations, such as hopper, chute and drum flow, and be applied to a range of geophysical and industrial scenarios. The majority of experiments take a three-dimensional (3D) approach and track the movement of individual grains or groups of grains via image analysis through the sidewalls, top and occasionally, interior of the flows [1–4]. The results can reveal flow velocity profiles, solid concentration and other quantities that are important to establishing models of granular flow behaviour [1, 5, 6]. Force transmission coupled with movement between individual particles is also of interest, although it is not yet possible to measure the microscopic behaviour of individual particles within a 3D granular flow. It is well known that particulate systems show a heterogeneous distribution of interparticle forces which carries most of the external load by way of force chains. The evolution of force chain microstructure is thought to play a critical role in determining the mechanical properties of granular assemblies but the physical mechanisms linking the mesoscopic structures represented by force chains to the macro scale bulk properties are not yet well understood.

The photoelastic technique has proved to be highly effective in experimental granular material research to gain a comprehensive insight of the particle scale dynamics and kinematics for two-dimensional (2D) granular systems, since it is possible to accurately extract information about the forces at grain-grain contacts when circular disks are used [7–9]. When visualised through stress-induced birefringence within an ensemble of photoelastic disks, the complex force networks appear highly ramified and can undergo rapid changes in their morphology throughout the material's loading history. The local stress produces, in each photoelastic disk, a pattern of alternating bright and dark fringes from which it is possible to obtain normal and tangential force components for each contact by solving the full inverse photoelastic problem, as described in Section 2. The estimation of the interparticle forces may be achieved through algorithms that have different implementations, however, they all require the positions of particles and contacts to be accurately determined [10–12].

In static or quasi static experiments, **disks detection** can be achieved with sufficient accuracy using the Hough transform [13]. Recently, technological improvements in high speed imaging has enabled researchers to perform photoelastic tests of very fast phenomena such as avalanching flows in which transparent particle disks of comparatively high compliance are arranged in a two dimensional sloping channel to ensure planar movement [14–16]. However, images of grain flow taken at high speed produce challenges in determining the precise positions of particles with standard approaches [17] .

In this work we have investigated the adoption of an Artificial Intelligence (AI) based methodology to address the task of identifying with high accuracy and reliability the disks embedded within challenging photoelastic images, characterised by bright force chains, noise, artefacts and geometric errors (e.g. parallax errors). More specifically, we have explored Deep Learning solutions to develop an effective technique to perform precise *image segmentation* to disentangle the disks from the environment in which they are embedded, and to outline them exactly. Deep Learning has proved to be effective for tasks relating to image and signal processing, computer vision, text analysis, and speech recognition [18–22] among others. It is a branch of Machine Learning that has become increasingly popular in the last decade, thanks to two concurrent factors: the availability of sufficient computing power to cope with complex, multilayered neural networks, and the availability of enough data to perform the training. Recently, it has also been adopted for several applications in the field of digital photoelasticity imaging, such as [23] and [24].

Various examples of image segmentation methods based on Deep Learning techniques are available. For an exhaustive up-to-date review we refer to [25]. For our photoelastic data, we have adopted a modified version of the U-Net architecture [26], customised to the problem under investigation. The description of the implemented Deep Learning methodology is given in Section 3.1. In Section 3.2, we introduce the training dataset, which is composed by a sufficiently large number of training images, that account for both the original photoelastic images (representing the input to the network), and the corresponding *reference* counterparts (images segmented using some alternative procedure), to compare with the results of processing performed by the network. The realisation of such reference images requires a significant effort, comprising manual processing of the data.

In Section 4, the detailed description of the procedure followed to select the most suitable network architecture for our photoelastic data is given. The resulting optimal configuration is then presented. The performance of the architecture is finally optimised by tuning the main model hyperparameters.

The results obtained by applying the network to different image sets are presented in Section 5. The accuracy of the network is evaluated, the interparticle forces are detected and the computational performance is discussed.

Finally, a summary of the key results together with the conclusions are drawn in Section 6.

2 The Photoelastic Images

Photoelastic materials rotate the polarization of transmitted light by a known amount that depends on the local stress, producing a visual pattern of alternating bright and dark bands/fringes

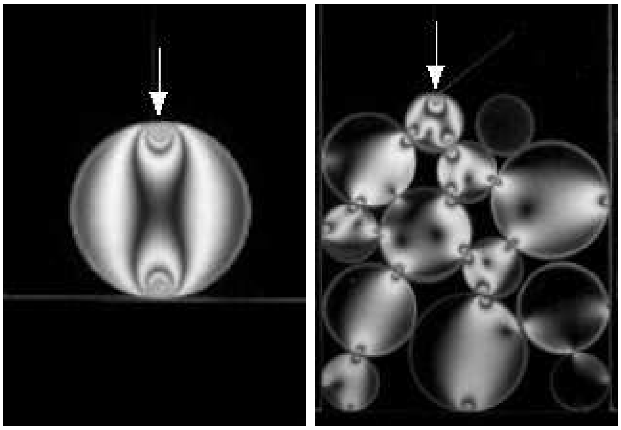

Figure 1 An example of the observed stress pattern for a single disk loaded from the top (left panel). An experimental image of stress patterns in a collection of photoelastic disks: particles that appear brighter and with more fringes are those experiencing larger forces (right panel). The arrows show the direction of the applied force.

with the number of bands increasing monotonically with the force at the contact, as shown in Figure 1. The resulting pattern depends on various factors such as the orientation and type of polarizers, the shape and thickness of the materials and the applied stress [27]. In previous research, semi-quantitative investigations, without knowledge of the vector contact forces between the particles, were performed on photoelastic images of circular disks [28–32] by computing the light intensity gradient-squared (also known as the G^2 method), to obtain measurements of global stress on particles [33].

If high-resolution input data are available, contact forces from disks can be extracted using photoelastic inverse methods. The equations for the image intensity due to a known set of particle positions and contact forces are known analytically, however, the inverse of the problem is not. Hence, several computational solutions have been created for the inverse problem, based on the fitting of the observed photoelastic pattern to the plane elasticity solution for the stresses inside a disk, treating the force components as fitting parameters [11, 12]. This is achieved in a two-step process: first, the positions of the particles are detected; then an estimate of the individual contact force magnitudes on each particle is calculated using an algorithm to minimise the difference between the experimentally observed pattern and the pattern generated from the reconstructed forces.

To extract contact forces in our experiments we used the open-source tool PeGS [12], available on GitHub[1]. The quality of the fringe-inversion for PeGS improves significantly with increased centroid accuracy. An accuracy $\leq 0.05d$, where d is the particle diameter, is required to successfully invert the fringe pattern.

In static or quasi-static experiments, the positions of the particles can be identified with high accuracy using the Hough transform approach. Usually for these experiments, two images for each experimental configuration are recorded: one taken in unpolarised light for detecting particle locations via Hough transform, and one taken in polarised light for measuring the forces on those particles.

For dynamic experiments like those discussed in this work, in order to perform photoelastic analysis on continuously avalanching flows, the particle positions have to be measured from a single image, since it is impossible to freeze their position during testing. To overcome this issue a possible approach could be to set up a reflection polariscope, a colour camera and two different sources of light. In this configuration, typically, a polarized, monochromatic light source provides the photoelastic image via a polarising filter placed on the camera, whereas the other non-polarised light source, which is still transmitted through the camera-mounted polarising filter, provides other measurements such as the particles positions. However this configuration is not desirable for high speed testing since colour cameras

[1] https://github.com/jekollmer/PEGS

have reduced light sensitivity and optical resolution compared to monochrome cameras. In addition, [34] demonstrated that colour filter arrays (CFA) and subsequently required interpolation procedures (demosaicking algorithms) introduce errors into photoelasticity studies when recording the colour information in digital images.

Therefore, we have adopted a set-up using a circular darkfield transmission polariscope, a monochromatic green LED light and a monochrome high speed camera in order to obtain the most suitable configuration for the analysis of our experimental granular flows.

2.1 Experimental Methodology

We generated two-dimensional free-surface granular flows by releasing photoelastic disks into a small scale flume made of transparent acrylic [15] inclined at an angle of 30° to the horizontal. The experimental set up is shown in Figure 2. The channel is 1600 mm long and 8 mm wide. A 450 mm long by 100 mm deep section has been cut out in the base of the channel to accommodate the particles composing the erodible bed. The particles used for the experiments consist of 6.4 mm thick, 1180 kg/m^3 density disks made from Vishay Precision Group's PSM-4 PhotoStress material. The photoelastic response of the disks is obtained by back illuminating the flume section, where the erodible bed is located, with a monochromatic green LED light passing through a pair of opposite circular polarizers, one mounted in front of the LED light panel and the other on the opposite side of the transparent chute. The erodible bed was made using a bidisperse population of photoelastic disks, most of which were 10 mm in diameter, randomly placed together with disks of 20 mm diameter. Before testing, a well-mixed population of approximately 400 disks with diameter of 10 mm, 16 mm and 20 mm at an area ratio of 2:1:1, were placed into a tank located at the upper end of the flume. These were then released by manually opening an acrylic gate at the base of the tank, so that the disks from the tank flowed over the erodible bed downslope.

The disks were lightly coated with baby powder to minimise disk-wall friction. Tests were recorded at the maximum available resolution (1280 x 800 pixels) at 2000 fps by a Miro Phantom monochrome high-speed camera equipped with a Nikor 50 mm lens. Exploiting the whole capacity of the on-camera data storage, it was possible to record footage of the experiments lasting approximately 4s, i.e. 8000 frames captured for each test, sufficient to record the entire process of entrainment/erosion of the particle bed due to the overriding granular avalanche. We set our frame rate to ensure that the evolution of the mesoscale features of the phenomenon were detected based on the Rayleigh time scale for force chains proposed by [35], which results as $t_R \sim 10^{-3}$ seconds.

Examples of the recorded images are shown in Figure 3. A video provided in the online supplementary material shows part of an experiment. The recorded images show a black background where the disks edges are partially visible due to the presence of a bright ring around the outer edge of the particle. This is caused by a small quantity of frozen-in stresses arising from the cutting procedure adopted for their production (although minimised, some residual stress is always found). In addition, refraction at the disc edges tends to increase this effect. The resulting outline of the particles, even if not desirable for an ideal estimate of forces from the photoelastic analysis, facilitates the **identification** of particle positions. The presence of force chains is evident, producing linear sequences of illuminated particles whose brightness depends on the intensities of the transmitted inter-granular forces.

From the images, disks can be outlined according to the following image processing procedure:

1. The input photoelastic image is "cleaned", removing as many features not belonging to the disk edges, e.g. the bright pixels of the force chains, as possible. This is achieved by a thresholding filter.
2. The resulting image is processed by the Hough Circle Transform plugin, part of the UCB Vision Sciences library on Fiji platform[2] ([36]) to detect circular edges and their centres. Centres and radii of the circles are calculated and used to compose a new image containing only the corresponding circles. Since the disc edges are not perfectly outlined in the thresholded images, the Hough score h, that defines the sensitivity of the disk detection procedure (the lower the score threshold, the more tolerant

[2] https://imagej.net/plugins/hough-circle-transform

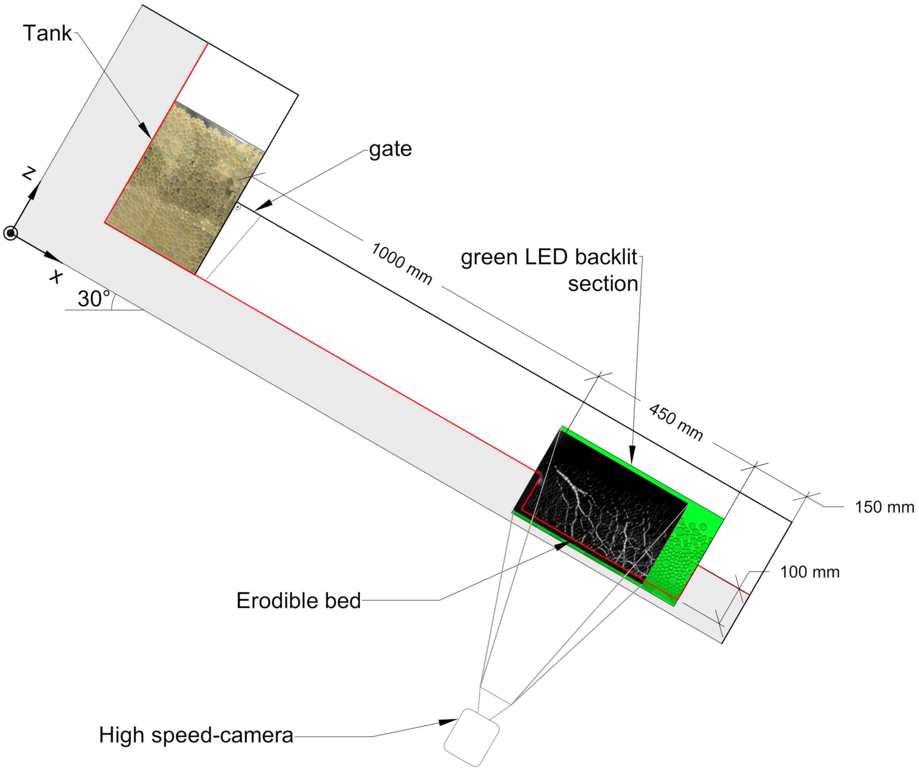

Figure 2 Schematic representation of the experimental set-up used to create 2D granular flows of photoelastic disks. The 2D avalanche flow is recorded by a high-speed camera as it passes over an erodible bed, passing between a pair of opposite circular polarizers backlit by a green LED light. This set up allows the visualisation of the force chain network.

the search will be of incomplete and/or imperfect circles) is set low to detect all the discs in the image ($h = 0.35$ for the smallest disks). However, such a setting has the drawback of leading to the introduction of a high number of spurious non existing disks (false positives).

3. The false positives have to be removed. This is achieved by adopting the following two-step approach : *i)* small circles embedded in larger circles are identified and removed using an automated procedure; *ii)* false positives resulting from the envelope of the edges of different disks are removed by visual inspection of each single image.
4. The final image is corrected by hand for circles that are not yet properly centred and sized due to the parallax error.

To apply to a large dataset, the resulting procedure is found to be error prone and highly time consuming, requiring continuous human intervention. A more robust and scalable procedure needs to be implemented in order to effectively process the photoelastic images and estimate the disk centres. A novel approach, based on the usage of Deep Learning, is therefore investigated in what follows.

3 Deep Learning Segmentation

The images under investigation, described in Section 2, have to be processed in order to precisely identify the centre of the disks. This results in a binary segmentation process, with the edge of the disks, characterised by variable brightness and parallax errors, to be recognised within a noisy background. For this purpose, we have investigated the usage of the *U-Net* architecture [26], an extension of a *fully convolutional network* originally developed for the segmentation of medical images.

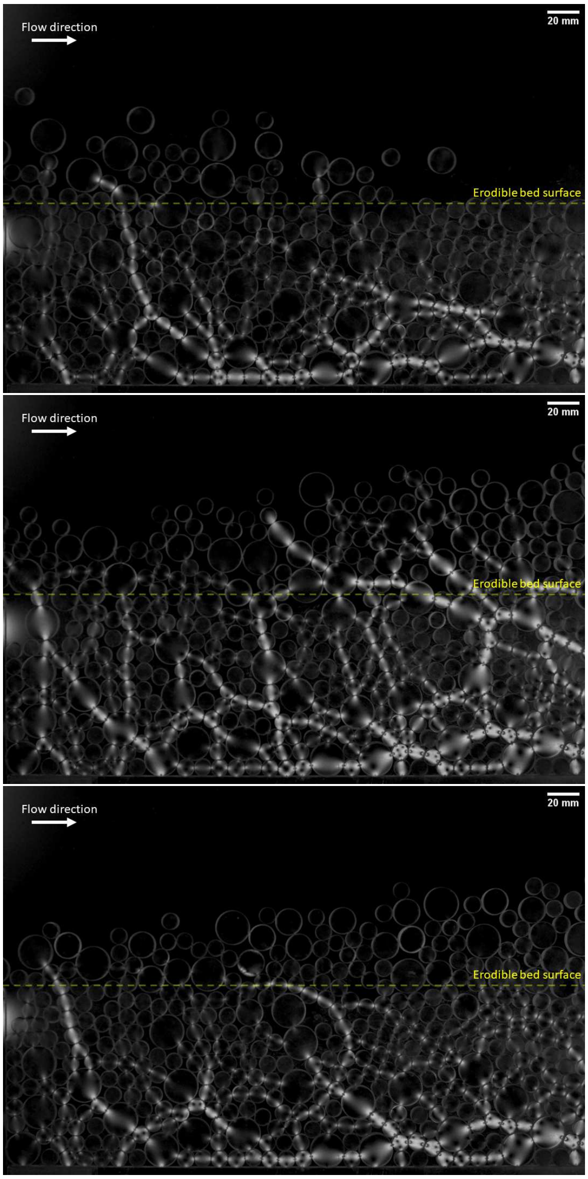

Figure 3 Example of the experimental images captured by the high-speed camera. The granular flow proceeds from left to right and the camera is tilted at 30° so that in the images the flow appears horizontal. The yellow dotted line indicates the top surface of the erodible bed. The top panel shows the arrival of the granular flow front; the centre panel shows the core of the flow; the bottom panel shows the tail of the flow.

3.1 The Convolutional Neural Network

We have adapted the original U-Net architecture to our problem. **Figure 4 shows an illustrative example of typical architecture for reference. The network consists of a contracting path and an expansive path. The contracting path is composed of an input layer, loading the input images, a downsampling convolutional network (composed by convolutional and pooling layers) which ends with a final level composed by convolutional layers. The expansive path starts from this deepest level, followed by an upsampling convolutional network, specular to the downsampling one. At each upsampling level, the feature maps are summed to the corresponding feature maps from the contracting path, in order to supply the spatial information to precisely localise the identified features. The final output layer returns the results. In our case, the results are represented by the segmented images, of the same size of the input ones, identifying the position of the edges of the disks.**

A 3×3 pixels window (receptive window) convolution operator is used, applying it starting from the input images:

$$s^f_{m,n} = \sum_{i,j=-1}^{1} w^f_{i,j} x_{m+i,n+j} + b^f , \tag{1}$$

where $x_{m,n}$ is the (m,n) pixel of the input image, $w^f_{i,j}$ and b^f are the weights and the biases of the convolutional kernel. The results of the convolution, the $s^f_{m,n}$ elements, compose the f-th *feature map*. A number of different feature maps are created starting from random initialisation of the weights and the biases. Non-linearity is introduced by further processing the $s^f_{m,n}$ elements by a proper activation function. We have adopted the ReLU activation function:

$$t^f_{m,n} = \max(0, s^f_{m,n}). \tag{2}$$

At this point, each feature map is downscaled through a max pooling function, which selects the maximum $t^f_{m,n}$ every $2{\times}2$ pool of feature map elements. The resulting maps are 1/4 the original size. The same convolution plus pooling procedure is repeated several times starting from a given set of feature maps, until the deepest layer is reached. The upsampling network reconstructs the

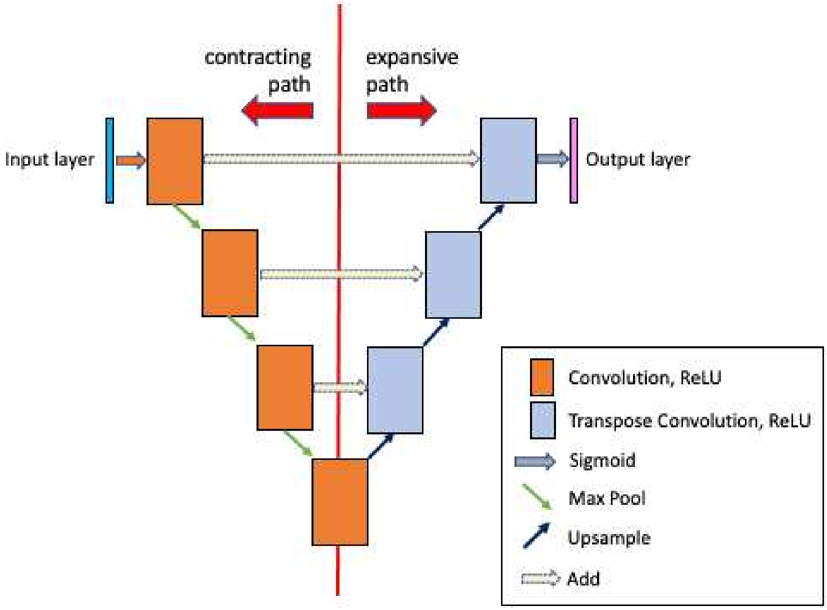

Figure 4 Illustrative example of a fully convolutional network architecture, developed on 4 levels (from 1, where also the input and output layers are present, to 4, the deepest level). It is composed by a contracting and an expansive path combining various kind of convolutional and pooling operators, processing, at each level, a given number of feature maps. Resolution (the size of the feature map) decreases (typically halves) moving from level N to level N+1.

full image through transpose convolutional and upsampling layers specular to the downsampling ones, starting from the deepest level. The result is returned by the output layer, which is, once more, a convolutional layer that uses a sigmoid activation function, commonly used in the case of binary classifiers:

$$p^f_{m,n} = \frac{1}{1 + e^{s^f_{m,n}}}. \tag{3}$$

The sigmoid function returns the probability $p^f_{m,n}$ of each single pixel of being or not part of the edge of a disk.

The parameters of the network (weights and biases) are optimised through an iterative procedure called *training*. **Two sets of images are needed for the training: the original input, i.e. the photoelastic images, and the reference images. The reference images are the results of the segmentation procedure obtained using the methodology described in Section 2.1, providing the "ground truth" data, i.e. a set of high accuracy disk edges data to compare to the results of the network.** The input images are processed by the network and compared to the corresponding reference images by means of a loss function, estimating the associated error. **We have adopted the categorical cross-entropy loss function, defined as:**

$$H(y_{m,n}, p_{m,n}) = -\sum_{f=1}^{N_c} y^f_{m,n} \log(p^f_{m,n}), \tag{4}$$

where $y_{m,n}$ is the pixel value in the reference image and N_c is the total number of classes to classify. For our application $N_c = 2$, i.e. a pixel belonging or not to a disk edge. The training aims at minimising H by correcting the network parameters through backpropagation of the cross-entropy error estimate, performed over successive sets of images. Batch normalisation and dropout have been used in order to improve the convergence of the method and to avoid overfitting.

Besides trainable parameters (weights and biases), additional network parameters, whose values are used to control the learning process and cannot be estimated via training, are referred as *hyperparameters*. For our network, the hyperparameters are:

- the learning rate μ, that regulates the step size of the iterative error minimisation procedure;
- the batch size B, that defines the number of images that are propagated through the network in one iteration step;
- the number of epochs E representing the number of times the training set is seen by the network during the training;
- the tile size T, that defines the 1D size of the tiles (the size of the tile is $T \times T$ pixels, see Section 2).

The network has been implemented using the Python programming language and exploiting the Keras software package ([37]), distributed as part of the Tensorflow framework ([38]), version 2.3.0. The resulting software can run on any computing platform, from standard multicore CPUs, available on a personal workstation, to high-end supercomputers, with computational performance changing according to the adopted architecture. In order to speed up the computation we have exploited the GPU implementation of Tensorflow. Furthermore we used distributed computing, based on the adoption of the MPI4Py library[3] ([39]) to run the various combinations of the hyperparameters of the model in parallel and identify the most accurate setup.

Training and tests were run on the Marconi100 (M100) High Performance Computing (HPC) system[4] available at the CINECA Italian Supercomputing centre. The M100 system consists of 980 computing dual socket nodes with 2x16 cores IBM POWER9 AC922 at 2.6 GHz and 256 GB memory per node. Each node is equipped also with four NVIDIA Volta V100 GPUs interconnected through Nvlink 2.0. The network is a Mellanox IB EDR DragonFly++ and 8 PB of GPFS storage are available in the scratch filesystem.

3.2 Training, Validation and Testing Datasets

The first step toward the usage of Deep Learning, is the creation of the *training* and *validation* datasets, extracting a sub-sample of the recorded images from our experiments. The former is used to train the network for the specific problem, the latter to verify the accuracy of the results. The validation data does not have to be used for the training. This is a a critical step, since each image belonging to the two datasets has to be supplied with a corresponding *reference* image, representing, with the highest possible accuracy, the true distribution of disks.

The procedure adopted to create the reference images is that described in Section 2.1. Due to the difficulties in its usage, we generated a relatively limited number of images, equal to 100.

With these images we have performed a first training of the network. The 1280×800 pixels images of the training datasets have been further divided into square tiles that become the actual training set of the network. Tile size is chosen in order to be the smallest possible still representative of the features to be identified (the disks of different sizes). In this way we maximise the size of the training sets without losing the significance of each single input image, reducing at the same time, the memory footprint of the network. Tiles in the lower part of the image (y coordinate > 512 pixels) have been excluded in order to avoid overfitting problems deriving from the fact that disks at the bottom of the erodible bed barely move, representing static, repetitive patterns.

An effective tile size for our images was found to be 256×256 pixels. The only relevant drawback of the tiling procedure is represented by possible small mismatches at the tile boundaries, which has been solved by overlapping the tiles by half their size.

Once the first training is done on the first 100 images, we use the resulting network to perform the segmentation of a further 200 images. Although the network is still trained with a limited number of training images and the results are not highly accurate, they are already good enough to ease the adoption of the procedure described in Section 2.1. At the end of this procedure, the training and validation datasets are composed of 2200 tiles overall.

An additional set of 100 photoelastic images, unseen by the network, compose the *testing* dataset. These images are used for

[3] https://mpi4py.readthedocs.io/en/stable/index.html

[4] https://www.hpc.cineca.it/hardware/marconi100

the final performance tests of the trained network.

4 Set-up of the Network and Training

The training of the autoencoder consists of the following steps:

1. the training program loads the input parameters, the hyperparameters and sets up the network;
2. training images end corresponding reference images are read from JPEG files stored on disk;
3. the results are normalised so that each image has values between 0 and 1;
4. images are divided into tiles;
5. tiles are serialised to feed the network;
6. tile batches are offloaded to the GPU and there processed for the training;
7. once convergence is reached, the trained network is saved in the *Model* file.

For the testing, the trained network is loaded from the Model file by the evaluation program which performs the segmentation of test images selected from a dataset never "seen" before by the network. The segmented image is finally saved in a TIFF file.

4.1 Network Set-up and Optimisation

Different network architectures have been tested in order to identify the most effective for our photoelastic images. The architectures differs in their depths (number of levels) and in the number of feature maps per convolutional layer at each level. The tests have been performed using the following hyperparameter setup: $T = 256$, $\mu = 0.0001$, $B = 50$, $E = 100$, $D = 0.7$ (this choice is justified below). **The results have been evaluated qualitatively by visual inspection and quantitatively by the accuracy calculated as:**

$$\epsilon = 1 - \frac{1}{N} \sum_{p=1}^{N} |Y_p - P_p| \tag{5}$$

where P and Y are the segmented and the corresponding reference images respectively, and N is the number of pixels in each of the two images. An accuracy equal to 1, means a perfect matching between segmented and reference images.

We have run the network configurations presented in Table 1. In the table we report also the accuracy ϵ, calculated over the whole image, and the accuracy ϵ_r, calculated on the most challenging part of the image. This is represented by the vertical stripe including the last 256 pixels in the horizontal axis. This area is subjected to the accumulation of the baby powder used as lubricant for the disks to minimise their friction with the acrylic sidewalls of the channel. The powder increases the background noise.

The results show that the *diskNet5* network gives the most accurate results, both considering the whole image and the right band, with an accuracy around 0.9316 and 0.9, respectively.

In order to better understand the quality of the results and the differences between the results produced by the different models, we show in Figure 5 the segmented images obtained for different network configurations and the corresponding input photoelastic image. Only models diskNet1, diskNet5, and diskNet6 are presented, the others having intermediate characteristics. In all cases, all the noise, artefacts and the force chains are fully removed. The diskNet5 model is our best result, showing that all the disks are neatly identified as circles with correct size and distribution. Only one circle is missed, on the bottom-right edge of the image. In contrast, Model diskNet1 is our worst result. In the corresponding image, many disks have an irregular shape, sometimes far from being circular. Spurious artefacts are present at the left edge of the image. In the diskNet6 model, circles are well shaped, however in some cases, in particular at the right edges, they tend to be fuzzy or not fully traced. Furthermore, at the base, most of the disks are missing.

As a result of our analysis, the selected network model is diskNet5, whose architecture is sketched in Figure 6. It is composed by 4 levels both in the downscaling and in the upscaling sweeps. The uppermost level (level 1) of the downscaling path is made of the input layer and two convolutional layers with 32 feature maps each. For the upscaling path, the same typology of convolutional layers is present together with the output layer, returning

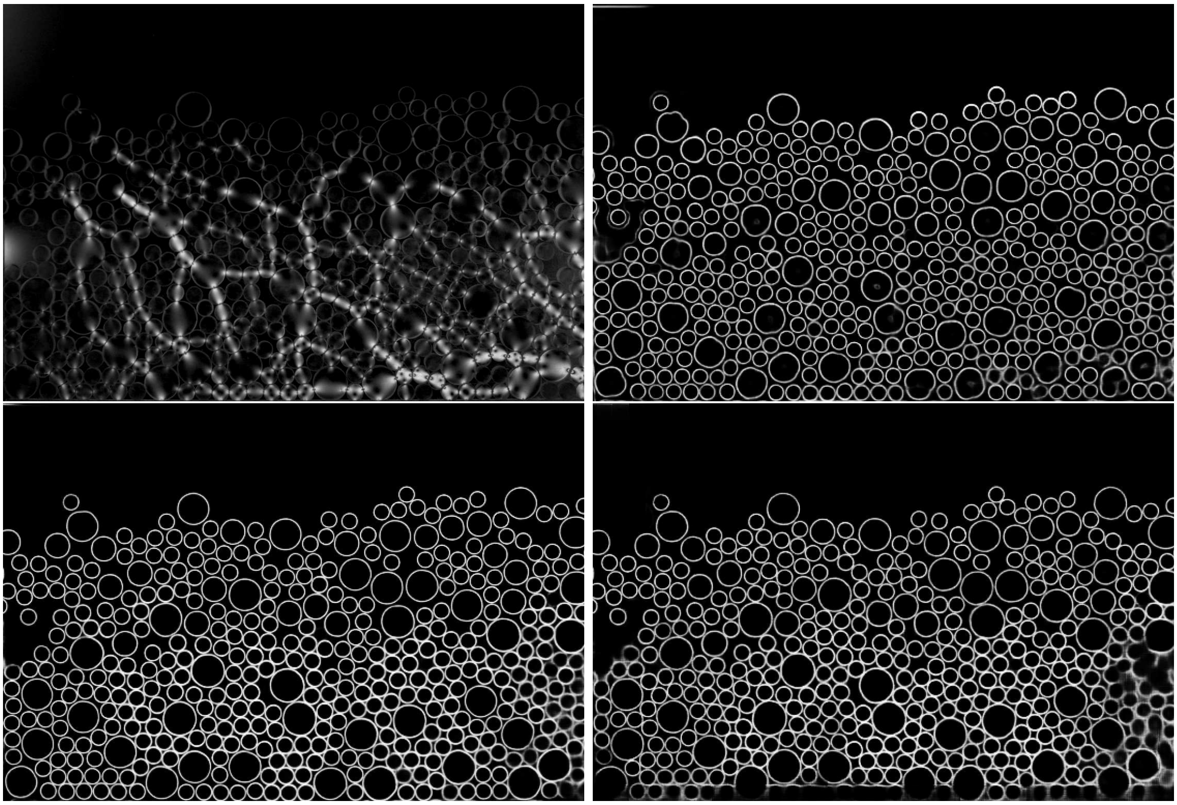

Figure 5 Segmented images obtained from different networks. For clarity only three of the models are shown: diskNet1 (top-right), diskNet5 (our best model, bottom-left), diskNet6 (bottom-right). The original photoelastic image is shown in the top-left panel. The granular flow proceeds from left to right and the camera is tilted at 30° so that in the images the flow appears horizontal.

Table 1 Tested architectural configurations for our network. The first column lists the network ids. The second column shows the depth D (i.e. the number of convolutional layers) of each configuration. The columns from 3 to 7 shows the number of feature maps at each layer. Finally, the last two columns shows the accuracy of the segmentation applied to the full images and to the last 256 pixels in the horizontal direction band.

Net. ID	D	Lev. 1	Lev. 2	Lev. 3	Lev. 4	Lev. 5	ϵ	ϵ_r
diskNet1	3	2×32	2×64	2×128	0	0	0.9057±0.0021	0.8803±0.0030
diskNet2	3	2×64	2×128	2×256	0	0	0.9308±0.0014	0.8945±0.0024
diskNet3	4	2×32	2×64	2×64	2×128	0	0.9283±0.0013	0.8991±0.0028
diskNet4	4	2×32	2×64	2×128	2×256	0	0.9278±0.0015	0.8953±0.0027
diskNet5	4	2×64	2×128	2×256	2×512	0	0.9316±0.0013	0.9000±0.0028
diskNet6	5	2×32	2×64	2×128	2×256	2×256	0.9182±0.0015	0.8874±0.0028

the feature map. The other three levels (levels 2, 3 and 4) in the contracting path are made of a max-pooling layer, rescaling down the size of the feature maps by a factor of two, and two convolutional layers, made of 64, 128 and 256 feature maps each, at increasing depth. Levels 2, 3 and 4 in the expanding path are made of the same number of convolutional layers and feature maps plus one upsampling layer. The deepest level is made of two convolutional layers with 256 feature maps each.

Based on diskNet5, we have optimised the network by studying the accuracy of the results as a

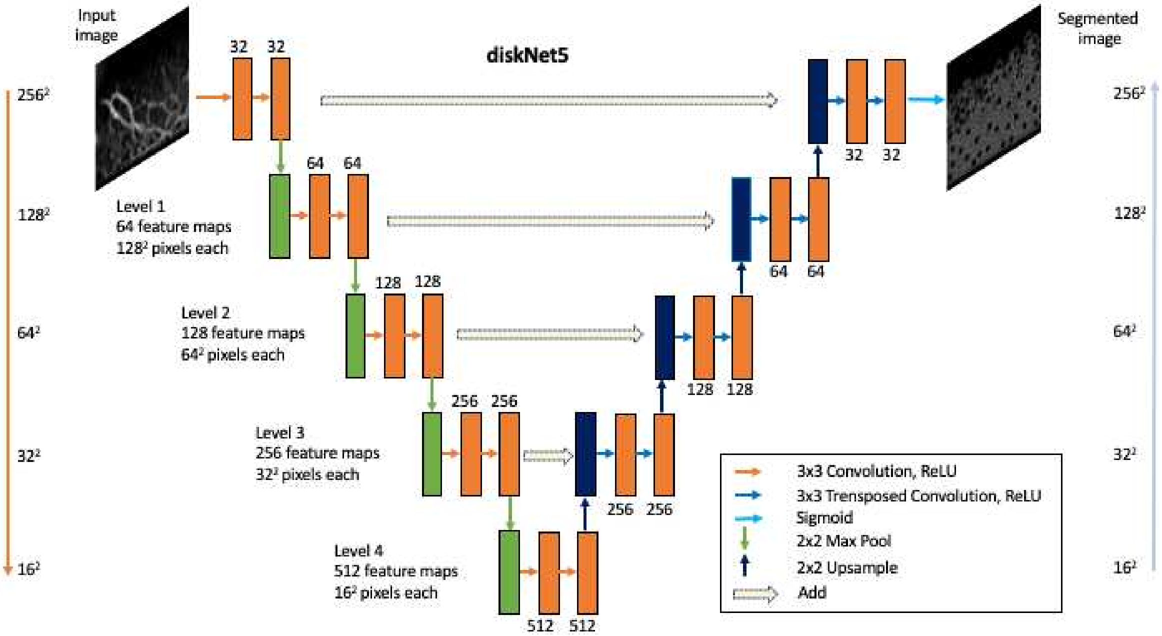

Figure 6 The diskNet5 architecture adopted for the segmentation of the photoelastic data under investigation. The network has 5 levels, at resolution between 256^2 and 16^2 pixels, two convolutional layers per level each with a different number of feature maps. The output layer has the same size of the input layer (256^2 pixels).

function of the hyperparameters. We have investigated all combinations of the following sets of parameters: $B \in [20, 50]$, $E \in [10, 50, 100, 150, 300]$, $\eta \in [0.001, 0.0001, 0.00005]$.

The results show how a batch size $B = 20$ always leads to results less accurate than the corresponding case with $B = 50$, which is then chosen as the optimal batch size. A sufficiently high number of epochs is key to the accuracy of the algorithm. The best result is obtained for 100 epochs, $\epsilon = 0.9316 \pm 0.0013$, while 10 epochs is ruled out (due to low accuracy, $\epsilon < 0.85$) and 50 epochs gives results slightly worse than the 100 epoch case. The tests with 150 epochs indicate that such a large number leads to overfitting. Finally, in all cases a learning rate of 0.001 leads to an accuracy < 0.90, while the highest accuracy is obtained for $\epsilon = 0.0001$. Therefore the selected configuration is $B = 50$, $E = 100$ and $\mu = 0.0001$.

It is worth noting that testing a variety of network architectures and the exploration of a wide hyperparameter space has been strongly facilitated by the MPI implementation of the code and the availability of a parallel computing architecture, which allows the distribution of different settings on different processors, running multiple models simultaneously.

5 Accuracy and Performance Analysis

We have investigated the performance of the diskNet5 network for different kinds of input datasets, resulting from different experiments, in order to explore a variety of system configurations. We have the objective to *i*) probe the capacity of the network to effectively perform the segmentation of the images, disentangling the disks from the complex environment in which they are embedded and identifying the correct number of disks without introducing false positives; *ii*) measure the accuracy of the method in estimating the centre of the disks; and *iii*) verify the correctness of the photoelastic patterns generated by the PeGS method using the diskNet5 segmentation results. Finally, we discuss the computational requirements and performance of the network.

The analysis of the images in the different tests highlighted a problem in the experimental set-up that slightly reduces the field of view

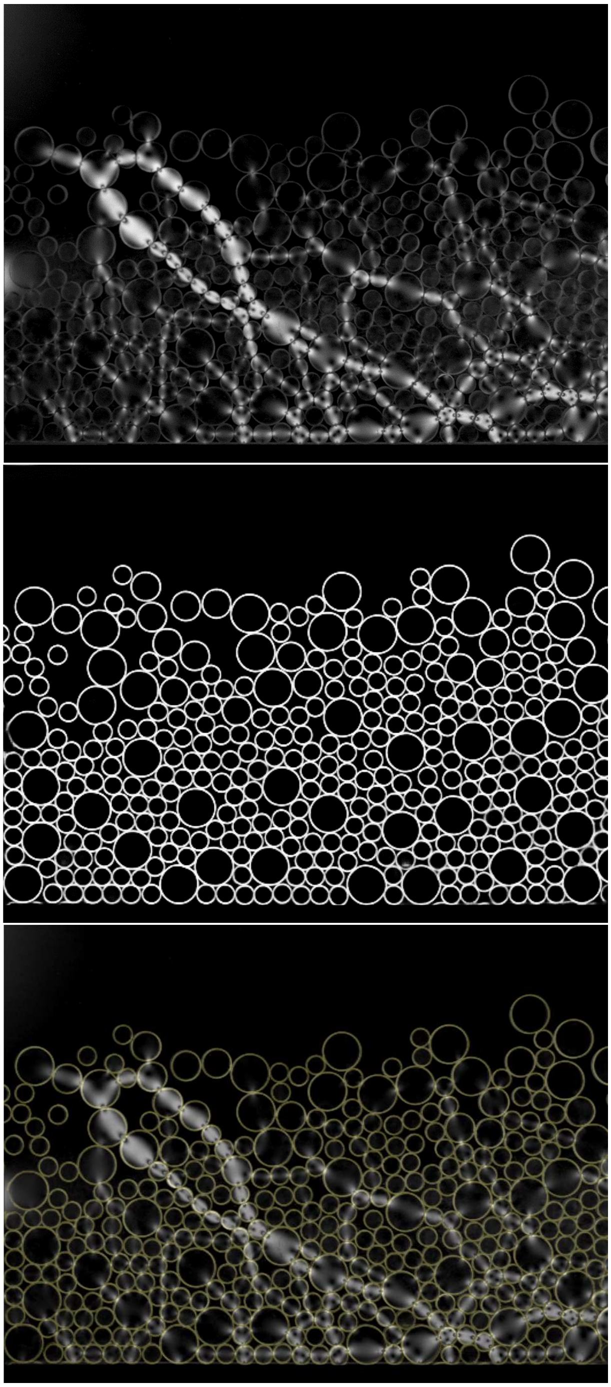

Figure 7 Comparison of an experimental photoelastic image (top panel, grey scale) with the corresponding segmented image (mid panel, binary). In the bottom panel the two images are superimposed, with the segmented circles highlighted in yellow. The granular flow proceeds from left to right and the camera is tilted at 30° so that in the images the flow appears horizontal.

available for data analysis. At the bottom-right corner some baby powder used to lubricate the disks accumulated as the disks in the erodible bed were offloaded from the cut out section, dragging them towards the downstream end of the apparatus. This increased the noise, making the data processing procedure challenging in that corner. Therefore, we have limited the analysis to the part of the images unaffected by the above problem (about 9/10 of the full image). This issue demonstrates how careful preparation of the experimental apparatus is key to the effective application of the processing methodology to an entire dataset, avoiding information loss.

The results show that the diskNet5 network is able to detect all the disks present in the image. An example is shown in Figure 7, where we have superimposed the result of segmentation (yellow circles) to the corresponding experimental, grey scale image. Infrequently, false positives have also been identified. They are related to specific patterns in the disk distribution that the network resolves as actual disks. An example is shown in Figure 8, where the envelope of the edges of the disks within a loop of disks is interpreted by the network as an additional disk. The frequency of the false positives has been estimated in our test data to be around 1 every 18 images, corresponding to about 1 every 6000 detected disks.

The accuracy in determining the coordinates of the centres is estimated according to the following procedure applied to the small disks, which are the most error prone in estimating the centroid:

1. Rectangular 256 × 256 tiles are cut out from the experimental images and from the corresponding segmented images randomly selected from the experimental tests. Three tiles, one on the left one on the centre and one the right, are extracted from each image.
2. The centres of the disks embedded in the diskNet5 segmented data tiles are calculated using the Hough Circle Transform to extract the list of the disks from the segmented images and estimate the coordinates of their centres. For the diskNet5 data however, the Hough score parameter can be set to $h = 0.7$, which avoids the introduction of any spurious false positives by the Hough algorithm itself (see Section 2.1).

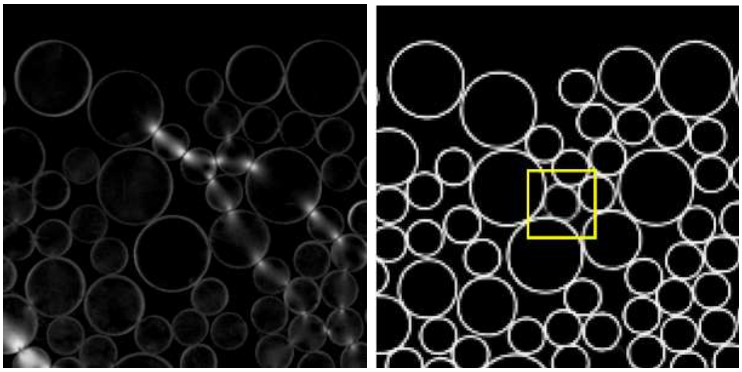

Figure 8 Example of one of the few false positives generated from the diskNet5 network. In the left panel a rectangular region extracted from an experimental image. An empty space forms among disks. In the right panel, the same region extracted from the segmented image is shown. A spurious disk appears, filling the empty space (yellow square).

3. The centres of the disks embedded in the experimental data tiles are estimated using a custom Python based software tool requiring manual identification of the disks.
4. The *centre error* is calculated as the difference between corresponding centre coordinates estimated from experimental and segmented data.

Figure 9 shows the distribution of the centre errors. As stated above, an accuracy $\leq 0.05d$ is required to successfully invert the fringe pattern using the PeGS software [12]. In the experiments, the diameter of the smallest disks is $d \sim 30$ pixels, corresponding to a maximum error of about 1.5 pixels. More than 85% of the centre positions are estimated with accuracy below this limit. The overall statistical error, defined as the average of the centre errors, is equal to 0.81 ± 0.54 pixels.

We have applied the PeGS method to images segmented using both diskNet5 and the semi-automatic procedure described in Section 2.1, in order to estimate the interparticle forces. Two examples of the obtained results are shown in Figure 10, where the synthetic force distributions obtained from the PeGS software are presented together with the corresponding original photoelastic images. The PeGS results accurately reproduce the actual force distribution for both segmentation procedures. However, the images generated by the semi-automatic approach are affected by a larger error in estimating the disk positions. This leads to missing a number of contacts between neighbouring disks, whose interparticle forces cannot be properly calculated by the software (highlighted in the yellow boxes in Figure 10). The same disk distributions are instead more precisely traced by diskNet5 and the corresponding photoelastic patterns are correctly estimated, confirming the effectiveness of the implemented segmentation methodology in image processing of photoelastic tests.

The computational requirements of the network, in terms of memory usage and computing time, are determined by its architecture, as well as by the image and the training dataset sizes. The memory requested by the trained network is determined by the the size of the receptive window, the number of hidden layers (depth of the network) and the number of feature maps. In the diskNet5 configuration, the network accounts for around 12300000 trainable parameters, corresponding to about 95 MB.

In the training phase, the required memory M grows substantially. It can be estimated as proportional to:

$$M = 8 \times B \times 4 \times \sum_{l=1}^{maxlev} \left(\frac{T}{l} C_l \right)^2 \text{GB}, \qquad (6)$$

where C_l is the 1D size of the feature maps for each level of the network and *maxlev* is the number of levels of the network. For the diskNet5 architecture, with input images (tiles) with $T = 256$ pixels (each pixel mapping to a 8 bytes quantity), the biggest possible batch size is $B = 50$, which leads to $M \approx 6$ GB. Bigger batch sizes exhaust the GPU memory and cannot be run.

The computing time of the training phases (the most time consuming part) depends on the specific

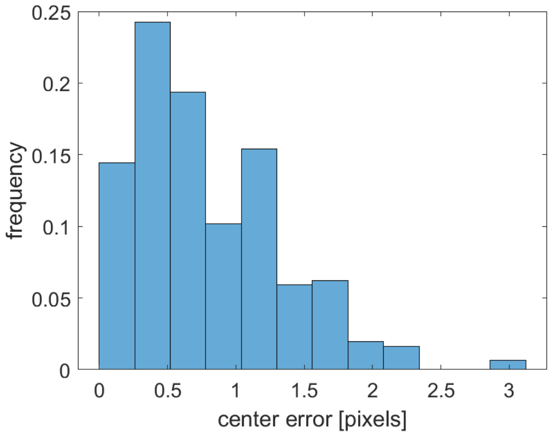

Figure 9 Distribution of the estimated error of the small disks centre positions obtained with the model diskNet5

Table 2 Approximate wall-clock times needed to perform various tasks with diskNet5. Image resolution is 1280×800 pixels, the training set is composed by 2000 images and the tile resolution is 256×256 pixels.

Task	sec.
Training time per epoch	19
Data load and preprocessing	24
Process time per tile	0.009
Segmentation time per image	0.3

adopted set-up, and in particular, on the number of epochs. The wall-clock time scales linearly with this parameter: the time to process a 2000 images training set is 226 seconds for 10 epochs, 983 seconds for 50 epochs and 1933 seconds for 100 epochs. The average time to process a single 256×256 pixels tile is ≈ 9 milliseconds. The time required to train the network also depends linearly on the training set size. Additional time is necessary to load data in memory from JPEG files and to perform the tiling of the images, as described in Section 3.2. The preprocessing time is proportional to the training set size and corresponds to ∼ 24 seconds for our training data.

Once trained, the network is used to process the test images. The time to perform the segmentation of our prototype 1280×800 pixels image is about 0.3 seconds on the adopted computing system. The time required to process the full results of a single experiment, composed by 4000 images, is ∼ 1200 seconds (about 20 minutes). **A summary of the characteristic computational times is presented in Table 2.**

6 Conclusions

We have studied the usage of a fully convolutional neural network approach to accomplish the segmentation of photoelastic images, removing noise, artefacts and other sources of confusion (e.g. force chains), identifying the disks embedded in the system with high accuracy. This is key to apply the methodologies necessary to the analysis of the photoelastic response of each single disk, which requires an estimate of the disk centre position with the highest possible precision.

For our data, the adoption of traditional approaches, like image filtering plus Hough transform to identify circular edges, proved to have a limited effectiveness. Human supervision is continuously required and manual correction of the results is necessary in order to eliminate frequent false positives, to add missing disks or to correct for clearly wrong disk positions. This leads to an error prone and time consuming procedure that cannot scale to large numbers of complex images, as those produced by experiments on dynamic systems involving particles moving at high speed, such as the modelling of rapid granular flows.

Once trained, instead, the convolutional neural network is fully automated, it produces accurate results and it is fast, efficiently exploiting HPC solutions. The enabling of the code to a fully data distributed HPC implementation (in progress) will also support larger cases, currently ruled out by the limit imposed by the size of the single GPU memory.

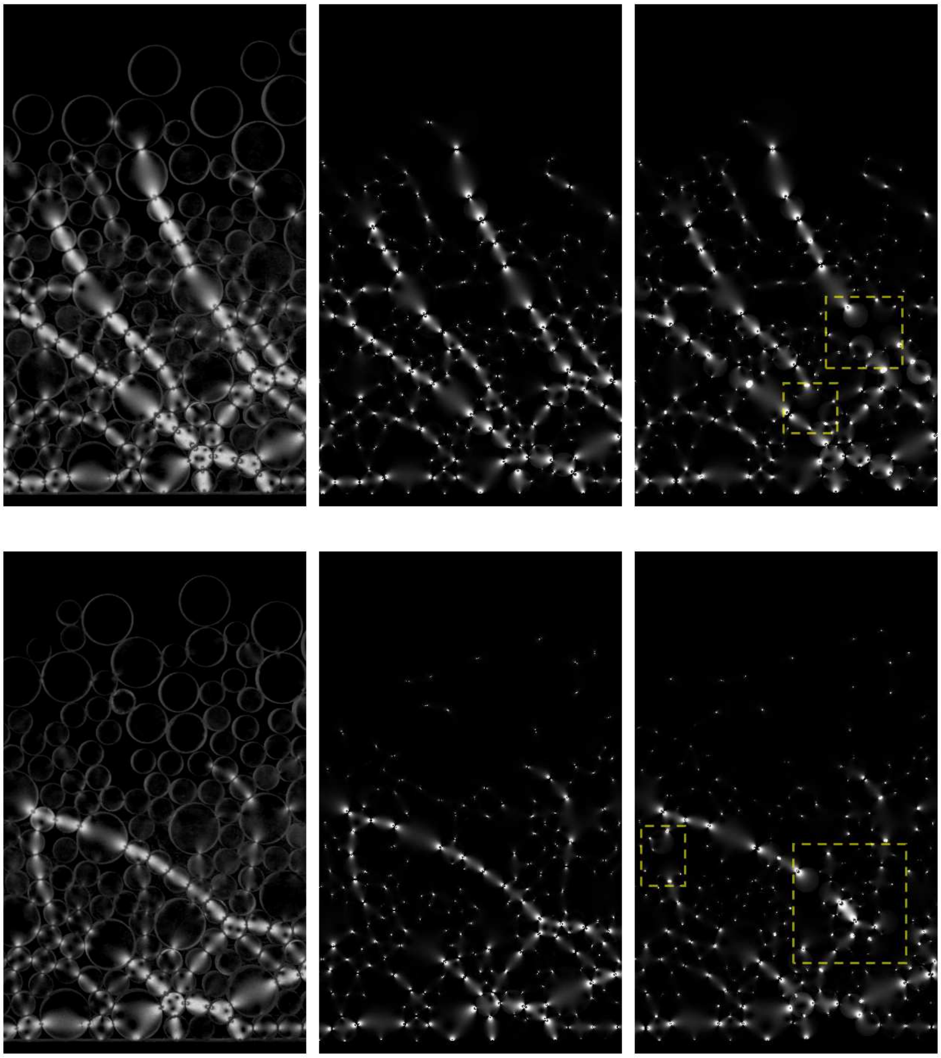

Figure 10 Example of the experimental photoelastic patterns (right panel) and the generated synthetic force chain network using PeGS. In the center panel the force network obtained with the disk centroids estimated with the model diskNet5; in the right panel the same region resulting from the centroids estimated via traditional approach combining preprocessing of the image and human supervision of the results. The main differences are located within the yellow squares.

Our main achievements can be summarised as follows:

- A Deep Learning approach has been developed and its performance extensively studied on experimental photoelastic images to verify its effectiveness in performing their segmentation identifying embedded disks.
- The method has proved to efficiently accomplish the segmentation task with an overall accuracy $\epsilon \sim 0.93$. All the disks embedded in the images are identified.
- The results are affected by the presence of few false positives, about 1 every 18 images or 6000 disks.
- **The proposed network, applied to experimental images with resolution of ~ 0.3 mm/pixel, produces segmented data in which the centres of the disks can be estimated using the Hough transform method [36] with an accuracy of 0.81 ± 0.54 pixels. This accuracy is below the upper limit of $0.05d$ required to successfully solve**

the inverse photoelastic problem by the PeGS software.

- The method supports full automation: once trained, it can be used seamlessly on the input data. No preprocessing is necessary on the input images. No human intervention, supervision or postprocessing is required. This strongly accelerates the data processing procedure.
- A relevant side result of our analysis is that careful laboratory testing, i.e. avoiding accumulation of lubricant on the sidewall of the experimental apparatus, is key to the successful application of the proposed methodology, without losing valuable information.

The adoption of AI approaches is being further investigated to fully address the solution of the photoelastic problem, combining the presented segmentation methodology to a Deep Learning solution capable of estimating the interparticle forces, as also discussed in [23]. This will be subject of forthcoming work.

Acknowledgements

The Deep Learning runs this work is based on have been produced the Marconi100 Supercomputer at CINECA (Bologna, Italy) in the framework of the ISCRA programme, IscrC_RadioGPU project. The photoelastic experiments were conducted at the University of Sheffield, UK, and were supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/M017427/1 "High speed granular debris flows: new paradigms and interactions in geomechanics".

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

[1] Midi, G.: On dense granular flows. European Physical Journal E **14**, 341–365 (2004). https://doi.org/10.1140/epje/i2003-10153-0

[2] Sanvitale, N., Bowman, E.T.: Visualization of dominant stress-transfer mechanisms in experimental debris flows of different particle-size distribution. Canadian Geotechnical Journal **54**(2), 258–269 (2017). https://doi.org/10.1139/cgj-2015-0532

[3] Kaitna, R., Dietrich, W.E., Hsu, L.: Surface slopes, velocity profiles and fluid pressure in coarse-grained debris flows saturated with water and mud. Journal of Fluid Mechanics **741**, 377–403 (2014). https://doi.org/10.1017/jfm.2013.675

[4] Trewhela, T., Ancey, C., Gray, J.M.N.T.: An experimental scaling law for particle-size segregation in dense granular flows. Journal of Fluid Mechanics **916**, 55 (2021). https://doi.org/10.1017/jfm.2021.227

[5] Savage, S.B.: The mechanics of rapid granular flows. Advances in Applied Mechanics, vol. 24, pp. 289–366. Elsevier (1984). https://doi.org/10.1016/S0065-2156(08)70047-4. https://www.sciencedirect.com/science/article/pii/S0065215608700474

[6] Jop, P., Forterre, Y., Pouliquen, O.: A constitutive law for dense granular flows. Nature **441**(7094), 727–730 (2006) arXiv:cond-mat/0612110 [cond-mat.soft]. https://doi.org/10.1038/nature04801

[7] Coulais, C., Seguin, A., Dauchot, O.: Shear modulus and dilatancy softening in granular packings above jamming. Phys. Rev. Lett. **113**, 198001 (2014). https://doi.org/10.1103/PhysRevLett.113.198001

[8] Zhang, J., Majmudar, T.S., Tordesillas, A., Behringer, R.P.: Statistical Properties of a 2D Granular Material Subjected to Cyclic Shear. arXiv e-prints, 0906–2416 (2009) arXiv:0906.2416 [cond-mat.soft]

[9] Abed Zadeh, A., Barés, J., Behringer, R.P.: Crackling to periodic dynamics in granular media. Phys. Rev. E **99**(4), 040901 (2019) arXiv:1803.06028 [cond-mat.soft]. https://doi.org/10.1103/PhysRevE.99.040901

[10] Puckett, J.G.: State Variables in Granular Materials: an Investigation of Volume and Stress Fluctuations. PhD thesis, North Carolina State University (January 2012)

[11] Majmudar, T.S., Bhehringer, R.P.: Contact force measurements and stress-induced anisotropy in granular materials. Nature **435**, 1079–1082 (2005)

[12] Daniels, K.E., Kollmer, J.E., Puckett, J.G.: Photoelastic force measurements in granular materials. Review of Scientific Instruments **88**(5), 051808 (2017). https://doi.org/10.1063/1.4983049

[13] Abed Zadeh, A., Barés, J., Brzinski, T.A., Daniels, K.E., Dijksman, J., Docquier, N., Everitt, H., Kollmer, J.E., Lantsoght, O., Wang, D., Workamp, M., Zhao, Y., Zheng, H.: Enlightening force chains: a review of photoelasticimetry in granular matter. arXiv e-prints, 1902–11213 (2019) arXiv:1902.11213 [cond-mat.soft]

[14] Estep, J., Dufek, J.: Substrate effects from force chain dynamics in dense granular flows. Journal of geophysical research **117**(F1) (2012)

[15] Sanvitale, N., Williams, G.B., Bowman, E.T.: Experimental investigation of granular flow erosion via photoelastic method. In: Proceedings of the XVII ECSMGE-2019 (2019). https://www.issmge.org/publications/publication/experimental-investigation-of-granular-flow-erosion-via-photoelastic-method

[16] Thomas, A.L., Vriend, N.M.: Photoelastic study of dense granular free-surface flows. Phys. Rev. E **100**, 012902 (2019). https://doi.org/10.1103/PhysRevE.100.012902

[17] Thomas, A.L.: Photoelastic study of dense granular free-surface flow rheology and size segregation. PhD thesis, University of Cambridge (2020)

[18] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791

[19] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc., ??? (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[20] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)

[21] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015). http://arxiv.org/abs/1409.4842

[22] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778. IEEE Computer Society, ??? (2016)

[23] Sergazinov, R., Kramar, M.: Machine learning approach to force reconstruction in photoelastic materials. arXiv e-prints, 2010–01163 (2020) arXiv:2010.01163 [cs.CV]

[24] Toro, H.F., León, J.B.D., Restrepo-Martínez, A., Bedoya, J.W.B.: Fringe patterns recognition in digital photoelasticity images using texture features and multispectral wavelength analysis. Optical Engineering **57**(9), 1–15 (2018). https://doi.org/10.1117/1.OE.57.9.093105

[25] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image Segmentation Using Deep Learning: A Survey. arXiv e-prints, 2001–05566 (2020) arXiv:2001.05566 [cs.CV]

[26] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (2015)

[27] Phillips, J.W.: Photoelasticity. University of Illinois at Urbana-Champaign, 6–2662 (1998)

[28] Geng, J., Howell, D., Longhi, E., Behringer, R.P., Reydellet, G., Vanel, L., Clément, E., Luding, S.: Footprints in sand: The response of a granular material to local perturbations. Phys. Rev. Lett. **87**, 035506 (2001). https://doi.org/10.1103/PhysRevLett.87.035506

[29] Clark, A.H., Kondic, L., Behringer, R.P.: Particle scale dynamics in granular impact. Phys. Rev. Lett. **109**, 238302 (2012). https://doi.org/10.1103/PhysRevLett.109.238302

[30] Zheng, H., Dijksman, J.A., Behringer, R.P.: Shear jamming in granular experiments without basal friction **107**(3), 34005 (2014). https://doi.org/10.1209/0295-5075/107/34005

[31] Cox, M., Wang, D., Barés, J., Behringer, R.P.: Self-organized magnetic particles to tune the mechanical behavior of a granular system **115**(6), 64003 (2016). https://doi.org/10.1209/0295-5075/115/64003

[32] Zhao, Y., Zheng, H., Wang, D., Wang, M., Behringer, R.P.: Particle scale force sensor based on intensity gradient method in granular photoelastic experiments **21**(2), 023009 (2019). https://doi.org/10.1088/1367-2630/ab05e7

[33] Howell, D., Behringer, R.P., Veje, C.: Stress fluctuations in a 2d granular couette experiment: A continuous transition. Phys. Rev. Lett. **82**, 5241–5244 (1999). https://doi.org/10.1103/PhysRevLett.82.5241

[34] Briñez-de León, J.C., Restrepo-Martínez, A., Branch-Bedoya, J.W.: Computational analysis of Bayer colour filter arrays and demosaicking algorithms in digital photoelasticity. Optics and Lasers in Engineering **122**, 195–208 (2019). https://doi.org/10.1016/j.optlaseng.2019.06.004

[35] Sun, Q., Wang, G.: Mechanics of Granular Matter. WIT Press, (2013)

[36] Benjamin Smith: Hough Circle Transform. https://imagej.net/plugins/hough-circle-transform#introduction

[37] Chollet, F., et al.: Keras. https://keras.io (2015)

[38] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Paper and Software available from https://www.tensorflow.org/ (2015)

[39] Dalcín, L., Paz, R., Storti, M.: Mpi for python. Journal of Parallel and Distributed Computing **65**(9), 1108–1115 (2005). https://doi.org/10.1016/j.jpdc.2005.03.010