GridSort: Image-based Optical Bulk Material Sorting Using Convolutional LSTMs ⋆

Marcel Reith-Braun * **Albert Bauer** ** **Maximilian Staab** *
Florian Pfaff * **Georg Maier** *** **Robin Gruna** ***
Thomas Längle *** **Jürgen Beyerer** ***
Harald Kruggel-Emden ** **Uwe D. Hanebeck** *

* *Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany, (e-mail: marcel.reith-braun@kit.edu; maximilian.staab@student.kit.edu; pfaff@kit.edu; uwe.hanebeck@kit.edu).*
** *Chair of Mechanical Process Engineering and Solids Processing, Technische Universität Berlin, 10587 Berlin, Germany, (e-mail: a.bauer@tu-berlin.de; kruggel-emden@tu-berlin.de)*
*** *Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, 76131 Karlsruhe, Germany, (e-mail: georg.maier@iosb.fraunhofer.de; robin.gruna@iosb.fraunhofer.de; thomas.laengle@iosb.fraunhofer.de; juergen.beyerer@iosb.fraunhofer.de)*

Abstract: Optical sorters separate particles of different classes by first detecting them while they are transported, e.g., on a conveyor belt, and subsequently bursting out particles of undesired classes using compressed air nozzles. Currently, the most promising results are achieved by *predictive tracking*, a multitarget tracking approach based on extracted midpoints from area-scan camera images that analyzes the particles' motion and activates the nozzles accordingly. However, predictive tracking requires expert knowledge for setup and preceding object detection. Moreover, particle shapes are only considered implicitly, and the need to solve an association problem rises the computational complexity of the algorithm. In this paper, we present *GridSort*, an image-based approach that forecasts the scene at the nozzle array using a convolutional long short-term memory neural network and subsequently extracts nozzle activations, thus circumventing the aforementioned weaknesses. We show how GridSort can be trained in an unsupervised fashion and evaluate it using a coupled discrete element–computational fluid dynamics simulation of an optical sorter. We compare our method with predictive tracking in terms of sorting accuracy and demonstrate that it is an easy-to-apply alternative while achieving state-of-the-art results.

Keywords: Monitoring of product quality and control performance; Neural networks in process control; Machine learning methods and applications; Artificial intelligence in mining, minerals and metals; Process monitoring and fault diagnosis.

1. INTRODUCTION

Optical sorting is a machine-vision-based technology that separates a bulk material of particles of different classes into its components at high throughput rates of several tonnes per hour. Common applications of optical sorters are in the mineral (Robben and Wotruba (2019)) and food industry (Bruce et al. (2021)). There is a substantial and growing demand arising from the recycling sector since alleviating the worldwide issue of waste pollution is one of the most urgent problems of contemporary times. This is also increasingly reflected in new environmental laws. For example, the European Union's recycling strategy stipulates that recycling rates of 70 % for packaging waste by 2030 and 65 % for municipal waste by 2035 are to be achieved (Friedrich (2022)). In this context, optical sorting is regarded as the key player for developing a sustainable circular economy (Friedrich (2022)).

A typical optical sorter consists of a transport medium, such as a belt or a chute, that transports the bulk material to a nozzle array mounted after the transport medium, and a camera that analyzes the particle flow. At the nozzle array, particles of undesired classes are ejected from the particle stream with bursts of compressed air. An algorithm for optical sorting thus has to fulfill the tasks of recognizing particles of the reject class and activating the nozzles accordingly, taking, e.g., the particle position, movement, and shape into account.

⋆ The IGF project 20354 N of the research association Forschungs-Gesellschaft Verfahrens-Technik e.V. (GVT) was supported via the AiF in a program to promote the Industrial Community Research and Development (IGF) by the Federal Ministry for Economic Affairs and Climate Action on the basis of a resolution of the German Bundestag.

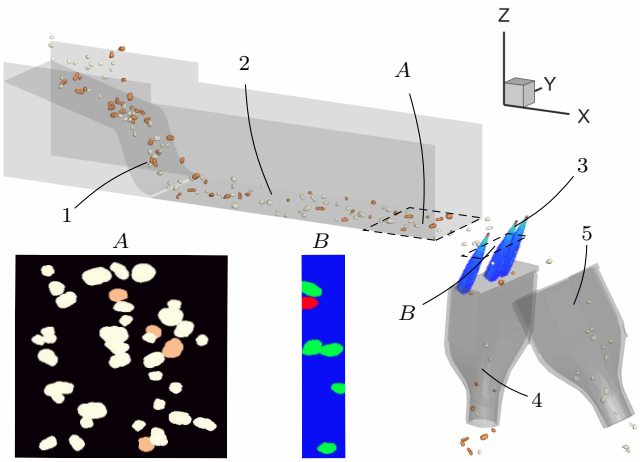

Fig. 1. Outline of GridSort and the DEM–CFD model of the sorter. The bulk solid is fed onto the feeding chute (1) and then transported by the conveyor belt (2) to the nozzles (3) while being analyzed by an area-scan camera with field of view A. The nozzles shoot particles of undesired classes into the reject hopper (4) using bursts of compressed air (depicted in blue), while accepted particles fall into the accept hopper (5). GridSort takes images A of the area-scan camera as input and forecasts an already segmented image B capturing the scene at the nozzle array. Based on this, nozzles (3) are activated in a postprocessing step.

Whereas current state-of-the-art sorters offered by the industry typically use a line-scan camera that captures a pixel row at the belt edge, in our previous works (Pfaff et al. (2015); Pfaff (2019)), we showed that sorting accuracy can be improved with the help of the *predictive tracking* paradigm. Here, a multitarget tracking (MTT) algorithm is employed on the particle midpoints extracted from images captured by an area-scan camera mounted above the belt. The extracted motion information is then used to precisely activate the nozzles. In predictive tracking, hard associations between measurements and predictions are determined in each time step using a global nearest neighbor approach. This is equivalent to solving a linear assignment problem, which, in predictive tracking, is either addressed by the Hungarian or the auction algorithm (see Pfaff (2019); Maier (2022)). For predicting and filtering particle states, multiple Kalman filters, one for each particle, using constant-velocity (CV) or constant-acceleration (CA) motion models are deployed. The prediction of the estimated particles' time of arrival and location at the nozzle array, also referred to as the prediction to the nozzle array, is then again accomplished with motion models inspired by physics, such as CV or CA models. To this end, the motion models use the estimated particle states from the Kalman filters. Whereas the results of predictive tracking show significant improvements compared with line-scan camera based sorters (Maier et al. (2021)), there are four major concerns about this approach:

- *Assumption of linear and independent particle motion:* Due to the use of independent, linear motion models for each particle, no complex motion patterns, no collisions with walls and other particles are covered.
- *Computational complexity grows exponentially with the number of particles*: Solving the association problem scales with the number of particles n in $\mathcal{O}(n^3)$, which constitutes a bottleneck for the usable frame rate. For example, Maier (2022) showed that even with GPU accelerated methods, this takes at least several milliseconds for $n > 400$.
- *Assumption of invariant particle shape projections*: In current implementations, only the last measured projection of the particle shape onto the belt (in the following referred to as the extent of the particle) is relevant for the nozzle activation. Thus, it is assumed that particle extents do not change compared with the last recorded measurement. However, this is a simplification since they can change due to rotations.
- *High complexity and sensitivity w.r.t. the choice of hyperparameters*: The sorting software needs to cope with various tasks (object detection and classification, MTT, prediction to the nozzle array), each of which represents a potential source of error. Each part relies on hyperparameters, which must be chosen carefully. Fine-tuning these parameters must be done in advance and requires expert knowledge, making the setup costly and the algorithm difficult to adapt to changing scenarios.

Therefore, we propose a novel approach called *GridSort* that directly forecasts the scene at the nozzle array as an image in an end-to-end fashion without the need to solve an association problem and extract the particles' centroids. The core of our proposed solution consists of a convolutional long short-term memory (ConvLSTM) based neural network that directly processes image sequences recorded by an area-scan camera and predicts an already segmented image as it would be captured at the position of the nozzle array P time steps ahead (see Fig. 1). Based on this intermediate result, the approach calculates nozzle activations in a minor postprocessing step, thereby automatically considering the particle extents. The use of a ConvLSTM network is inspired by similar tasks in environment prediction for autonomous driving. Here, one usually tries to predict the future occupancy of a grid representing the ego vehicle's surroundings based on input sequences of grid maps encoding, e.g., LiDAR or camera measurements (Dequaire et al. (2018); Mohajerin and Rohani (2019); Itkina et al. (2019); Lange et al. (2021)). In particular, we use the *grid predictor model* neural network architecture proposed by Schreiber et al. (2019) and adapt it to our scenario.

Our contributions are: We propose an approach for grid-based sorting using a ConvLSTM network. We demonstrate how the network can be trained in an unsupervised fashion, making use of the intermediate result, i.e., the network output, as a segmented image. We thus avoid parameter tuning as required for predictive tracking and laborious labeling of training data. We test the approach on a coupled discrete element–computational fluid dynamics (DEM–CFD) simulation of a sorter, demonstrating suitability for optical sorting. Thus, an alternative approach to predictive tracking is proposed, which for the first time allows forecasting particle extents across multiple time steps (see Fig. 2).

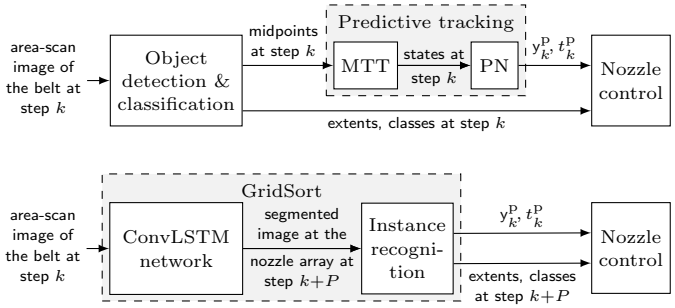

Fig. 2. Comparison of predictive tracking and GridSort. In predictive tracking, the particles' predicted time of arrival t_k^p and location y_k^p are calculated solely based on extracted midpoints from an object detector. The extracted particle extent and class information from time step k is included when activating the nozzles, even though it is possibly outdated because a particle may have changed its orientation when reaching the nozzle array. GridSort alleviates this by forecasting an already segmented image capturing the scene at the nozzle array P time steps ahead. PN abbreviates prediction to the nozzle array.

2. RELATED WORK

2.1 Predictive Tracking and DEM–CFD for Optical Sorting

Since its emergence, several improvements have been proposed for predictive tracking. Pfaff (2019) proposed to incorporate orientation estimation in the MTT to improve the association quality. Although in principle possible, the predicted particle orientations were not used for more precise activation of the nozzles, e.g., by taking into account predicted particle extents. Recent developments allow for more accurate tracking and prediction to the nozzle array using recurrent neural networks and multilayer perceptrons that replace the Kalman filters and the linear motion models (Pollithy et al. (2020); Thumm et al. (2022)).

DEM–CFD was used to simulate optical sorters in Pieper et al. (2018); Bauer et al. (2023) and showed high consistency with experiments on real sorters. Bauer et al. (2022) optimized the sorting setup for various belt velocities and showed that higher belt velocities can improve the accuracy under certain material feed conditions. DEM–CFD resolves contacts of particles with other particles, other components of the sorter, and the fluid field surrounding the particles in detail and in high temporal resolution. The discrete element method (DEM) relies on the fundamental laws of classical mechanics, i.e., Newton's law of motion and Euler's equation to infer the motion of particles from applied forces and torques. Concurrently, in computational fluid dynamics, the Navier–Stokes equations are solved numerically to account for fluid forces imposed by relative velocities between the air and the particles. Here, drag models are commonly used to estimate the fluid forces on the particles.

2.2 Grid-based Tracking and Prediction

Grid-based tracking and prediction approaches are popular in the autonomous robot and driving community, where they are employed, e.g., for motion prediction of road users in the vicinity. In grid-based tracking, the region of interest is usually discretized into rectangular grid cells, mostly using a two-dimensional birds-eye-view representation of the scene. Each cell value then encodes the occupancy or class of an object that covers the corresponding cell, thus avoiding the need to solve an association problem in MTT. On the top level, grid-based approaches can be divided into those using Bayes-filter-based methods encoding handcrafted motion models and those using deep neural networks learned on large data sets. Methods belonging to the first category usually deploy the Bayesian occupancy filter (Chen et al. (2006)) and variants thereof. Whereas such approaches allow for uncertainties in the prediction and proved to be useful for tracking and short-term prediction tasks, they usually suffer from blurring in the predictions because of inaccuracies of the hand-crafted motion models and accumulation of system noise on long-term prediction tasks. For those tasks, deep neural networks based on recurrent convolutions show promising results. The first publications on applying recurrent convolutions to occupancy grid tracking and prediction is the *deep tracking* article series by Dequaire et al. (2018). More recently, Schreiber et al. (2019) proposed an encoder–decoder architecture called *grid predictor model*, where the input sequence is first scaled down by an encoder convolutional neural network (CNN) with three layers to deep low-resolution feature maps and then fed into four ConvLSTM layers before being upscaled to the output shape by a four-layer non-recurrent upscaling CNN. Moreover, to prevent blurring of the output image, the authors proposed using skip connections, which may also contain recurrent ConvLSTMs, between the encoder and decoder layers. With this architecture, the grid predictor model was able to precisely predict future object motion in traffic scenarios up to 20 time steps ahead when fed with an input sequence of the same length. Mohajerin and Rohani (2019) applied a similar encoder–decoder ConvLSTM network as part of a difference learning approach on lidar grid maps. Another branch of related work focuses on the ConvLSTM network *PredNet* by Lotter et al. (2017), previously used for video prediction, that was applied for grid prediction by, e.g., Itkina et al. (2019) on the KITTI lidar dataset. Most recent, Lange et al. (2021) used attention mechanisms on the input and hidden state sequence of PredNet to highlight import features. They showed that one is thereby able to adequately predict up to 2.5 s into the future when fed with lidar measurements arriving at 10 Hz in challenging and highly dynamic scenarios.

3. GRID-BASED SORTING MODEL

Our GridSort approach takes an RGB input image directly from the area-scan camera at each time step k and, after preprocessing, feeds a sequence of the latest N input images $\mathbf{X}_{k-N+1}, \ldots, \mathbf{X}_k \in \mathbb{R}^{H_\mathrm{in} \times W_\mathrm{in} \times 3}$ into the ConvLSTM-based neural network predictor (see Fig. 3). The ConvLSTM network then outputs a prediction for the Pth time step in the future $\mathbf{Y}_{k+P}^* \in (0,1)^{H_\mathrm{pred} \times W_\mathrm{pred} \times 3}$ in the form of a segmented image in the vicinity of the nozzle array. Here, the components y_{ijl}^* of $\mathbf{Y}_{k+P}^*$, with $\sum_{l=1}^{3} y_{ijl}^* = 1$, encode the probability $\mathbb{P}(\mathbf{y}_{ij} = c_l)$ that the discrete random variable $\mathbf{y}_{ij}$, which describes the class membership of

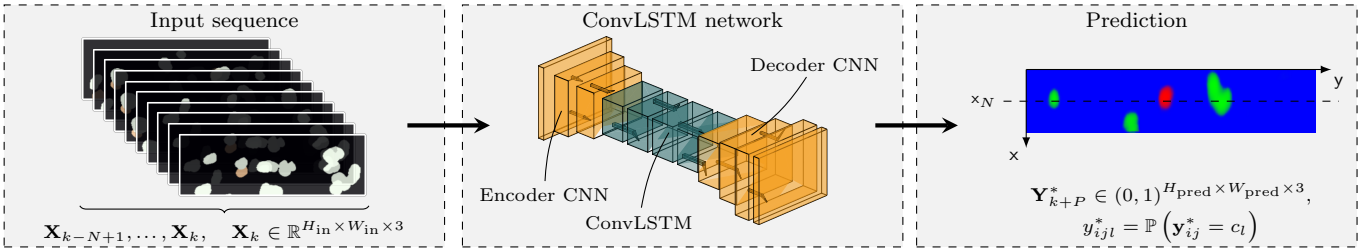

Fig. 3. Input sequence, ConvLSTM-based neural network architecture, and the forecasting for the scene at the nozzle array. The input sequence of the latest N (preprocessed) RGB images is fed into a ConvLSTM network in encoder–decoder architecture, where they are first compressed by an encoder CNN (left part) using three convolutional layers and then processed by four ConvLSTM layers (middle part). The last predicted feature map for $k+P$ is then upsampled by the decoder CNN, consisting of three layers (right part). A 1×1-convolutional layer with softmax activation subsequently yields the predicted image $\mathbf{Y}^*_{k+P}$ that encodes the class membership probabilities of each pixel.

the cell indexed ij, belongs to the class c_l indexed by $l \in \{1, 2, 3\}$. Note that the choice of three channels in the model output $\mathbf{Y}^*_{k+P}$ reflects that in a sorting task, we usually have three classes (eject particles, accept particles, and background). In the following, we first describe our neural network's architecture and then explain in detail how the nozzle activations are calculated from the network's predictions. [1]

3.1 Model Architecture

We build upon the grid predictor model proposed by Schreiber et al. (2019). Whereas its original version was used to predict a sequential output, we change the architecture to only output a single image at each model call. Therefore, we process the whole sequence until the last ConvLSTM layer, but then return only the last output in its output sequence. Additionally, we omit the skip connections between the encoder and decoder CNN, as our results have shown that the model capacity without skip connections is sufficient for our task. The final layer is a 1×1-convolutional layer with softmax activation, directly outputting the class probabilities $\mathbf{Y}^*_{k+P}$. We choose to work with an input image shape $H_\text{in} \times W_\text{in}$ for the ConvLSTM network of 132×408 px and a predicted image shape $H_\text{pred} \times W_\text{pred}$ of 88×408 px. The encoder CNN compresses the image to 11×34 px while increasing the number of channels to 64. In total, our model has 3 362 435 trainable parameters.

In the preprocessing, we resize the input image and standardize each of its channels to have a mean of zero and unit variance using feature-wise standardization with mean and variance determined on the whole data set. For postprocessing, we first change the size of the predicted image $\mathbf{Y}^*_{k+P}$ to a task-specific $H_\text{out} \times W_\text{out}$ and then transform the predicted class probabilities to one-hot-encoded predicted classes $\mathbf{Y}_{k+P} \in \{0, 1\}^{H_\text{out} \times W_\text{out} \times 3}$ by taking the argmax along the class dimension, thus using the most likely class. In the following, we refer to $\mathbf{Y}_{k+P}$ as the output of the model.

3.2 Extracting Nozzle Activations

For training and deployment, P and H_out are to be chosen such that P reflects the average particle travel time from the end of the area-scan camera field of view (FOV) to the nozzle array and H_out is broad enough to account for variations in the travel times as well as in the particle extents. From $\mathbf{Y}_{k+P}$, nozzle activations are calculated by first extracting the axis-aligned bounding boxes of the particles to be ejected. The y-coordinates of the bounding box edges are then directly used to determine which nozzles to activate. For calculation of the activation time t^p_k and duration Δt^p_k at time step k (both in time steps but as a floating number), the bounding box length in x-direction l^p and the x-coordinate of its center point c^p_x are transformed to a time by dividing by an average particle velocity v^avg_x (in pixels per time step) according to $t^\text{p}_k = \frac{\mathsf{x}_N - c^\text{p}_\mathsf{x}}{v^\text{avg}_\mathsf{x}} + P + k$ and $\Delta t^\text{p}_k = \frac{l^\text{p}}{v^\text{avg}_\mathsf{x}}$, where x_N denotes the pixel position of the nozzle array. Note that it is not possible here to use particle individual velocities as this would require associations over the last predicted frames, which we want to avoid.

4. DATA SETS

We use two distinct data sets for both training and evaluation. The first one consists of area-scan images recorded on the real-world lab-scale optical belt sorter described in Pfaff (2019); Maier (2022) whereas the second one stems from artificial data generated by the DEM–CFD model of the same optical sorter presented in Bauer et al. (2022). [2] The recorded material in both cases is construction and demolition waste, consisting of a mixture of brick and sand-lime brick.

4.1 Real Bulk Material Data Set

Recorded mass flows were 10, 15, and $20\,\text{g}\,\text{s}^{-1}$. The mixing ratios range from 50:50 to only one material each. The belt has a width of 140 mm and its velocity is approximately $1\,\text{m}\,\text{s}^{-1}$. Images were taken by an *Allied Vision Bonito CL-400 color* camera at a frame rate of 192 fps and a resolution of 2320×1726 px. The data set consists of 22 sequences, each of which correspond to one mass-flow-mixing-ratio pair. Each sequence comprises 7900 images, resulting in approximately 40 s of recorded material.

[1] Our source code is available at `https://github.com/KIT-ISAS/TrackSort_Grid-based/tree/IFACWC2023`

[2] Our data sets are available at `https://doi.org/10.5281/zenodo.7801882`, `https://doi.org/10.5281/zenodo.7782405`

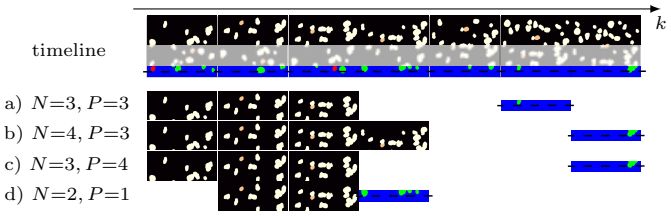

Fig. 4. Example illustration of the impact of the number of input images N and the time step difference to predict to P. The first row shows the images with division as described in Sec. 5, i.e., the upper part marks the region that is used as input to the ConvLSTM network, the shaded middle part represents the prediction gap that is invisible during training, and the lower part shows the segmented target at the vicinity of the virtual nozzle array (depicted by the dotted line). a) demonstrates an appropriate choice for N and P since the majority of the particles in the target images are visible in all three input images as illustrated by the displayed example sequence. The higher N in b) is of no value because the particles in the target images have not yet appeared in the first frames of the input sequences. In c), the higher P leads to less motion information the network can work with compared with a). In d), due to a too small N, information about the motion of the particles in the target images is not present at all in the input sequences.

4.2 DEM Data Set

The data set contains mid-points, radii, and classes of spheres that build sphere clusters, with each cluster modeling a particle. Input images are generated by rendering the sphere clusters in the FOV of the simulated area-scan camera. The particles have diameters from 4 to 8 mm. The simulated mass flows were $70\,\mathrm{g\,s^{-1}}$ for sand-lime brick and $30\,\mathrm{g\,s^{-1}}$ for brick. The belt velocity is approximately $1.1\,\mathrm{m\,s^{-1}}$. Data is recorded at 2000 fps and covers a simulated time period of 120 s.

5. UNSUPERVISED TRAINING PROCEDURE

In this section, we describe our unsupervised training procedure, which is based on the observation that labeled images can be obtained with only little effort using classical image segmentation methods. We train one model each for the real bulk material and the DEM data set.

5.1 Unsupervised Training Data Generation

In principle, one could directly use images from a camera mounted above the nozzle array, capturing the images for ground truth generation. However, as this requires additional hardware, we propose an alternative method that exploits the concept of a *virtual nozzle array*, as previously introduced by Pfaff et al. (2015). Here, for ground truth generation, a virtual line in the recorded area-scan image is considered as the nozzle array and the input image processed by the network is cropped accordingly while maintaining the distance between the end of the (now artificial) camera FOV and the (now virtual) nozzle array. Note that the concept of a virtual nozzle implies the assumption of only small deviations between the particle motion on the belt and during free flight (particle motion itself can still be very complex). However, it enables creating training and test data directly from the previously introduced data sets.

For this, we first divide the images into three parts (see Fig. 4). The upper part is used as the actual input image, whereas the lower part is used to create a labeled ground truth for training the ConvLSTM network. The intermediate part of the image is not used as it represents the prediction gap in the sorting system, i.e., the part after the camera FOV and in front of the nozzle array. The lower part is segmented into the three classes brick, sand-lime brick, and background using color-based segmentation in HSI color space as described in Maier et al. (2021); Maier (2022) in the case of real-world data and by directly exploiting the given class label in the case of DEM data. When deploying the trained model to a real sorting task, we restrict the camera FOV to the same height as for the training input image and shift it according to the distance between the beginning of the prediction gap and the virtual nozzle array, i.e., it is shifted so that the virtual and actual nozzle array positions match.

For both data sets we use a fraction of approximately 0.46, 0.37, and 1/6 of the original image height as input image, prediction gap, and target image, respectively. When assuming that the nozzle array x_N is located in the middle of the target image, this corresponds to a distance between the end of the area-scan camera FOV and x_N of approximately 78.4 mm for the real-world data set and 58.6 mm for the DEM data set. The latter corresponds to precisely the same value as in the DEM–CFD model by Bauer et al. (2022). For creating input and target images for the DEM data, we only use each 10th image, to achieve a frame rate comparable to that of the real-world data. An input sequence length $N = 10$ and $P = 10$ are applied for training, which lead to valid predictions for both data sets. Note that the choice of N and P heavily depends on the average particle velocity (see Fig. 4 for an explanation of the influence of N and P). Additionally, depending on the computation time of the algorithm and the time for nozzle activation, a small P may lead to a too small $t_k^{\mathsf{P}} - k$ that cannot be realized by the sorting system.

5.2 Training Settings

We train our models with categorical cross entropy loss, Adam optimizer with learning rate $\alpha = 0.0001$, and exponential learning rate decay with a decay rate of 0.8 per epoch. In total, the model is trained for 50 epochs. To account for the class imbalance in the data sets, during loss calculation, we weight the loss for each pixel with a weighting factor equal to three divided by the total number of pixels of this class in the target image. We use 80 % of the image data for training and another 10 % for validation and testing. As no overfitting could be observed so far, we refrain from using early stopping or dropout.

6. EVALUATION

Our evaluation consists of two parts. First, we evaluate the prediction accuracy of our ConvLSTM network on an offline

(a) Prediction on real-world data (b) Target on real-world data (c) Prediction on DEM data (d) Target on DEM data

Fig. 5. Qualitative comparison of prediction and ground truth of the proposed ConvLSTM network applied to the image forecasting and segmentation task. Red encodes sand-lime brick, yellow brick, and blue background.

prediction task using the two data sets presented previously. Second, we simulate a sorting task with GridSort using the DEM–CFD simulation model of Bauer et al. (2022) and compare it with predictive tracking.

6.1 Offline Prediction

Predicted images and corresponding targets are displayed in Fig. 5 for qualitative comparison. Predictions show good alignment with the ground truth, especially for the DEM data. For a quantitative evaluation, we calculate the pixel-wise confusion matrix of the three classes, as displayed in Tab. 1. The results show high accuracies for all classes with values between 89 % and 98 %. Again, the accuracy for DEM data is higher than for real-world data. We hypothesize that this is due to imprecise class segmentations in the ground truth images, since creating a pixel-accurate ground truth at the objects' borders is often an ambiguous task for real-world images. The mean inference time for one call of the ConvLSTM network is around 58 ms. [3]

6.2 DEM–CFD Sorting Simulation

For the sorting simulation, we use the same DEM–CFD model of the sorter, settings, and particle models as used to create the DEM data set. For deployment of the trained network, we apply the shifting and shortening of the camera FOV as described in Sec. 5. In order to account for mechanical and pneumatic inertia, a nozzle is blocked, i.e., it cannot be activated, for 3.5 ms before and after each activation. For v_x^{avg}, we use the value of the belt velocity, i.e., $1.1\,\mathrm{m\,s^{-1}}$.

We consider two different scenarios for evaluating the sorting accuracy. Scenario *S1* uses precisely the same mass flows as the DEM data set. Scenario *S2* has a much higher mass flow of $500\,\mathrm{g\,s^{-1}}$ and a mixing ratio of 70 % sand-lime brick and 30 % brick. The ConvLSTM network trained on the DEM data set was also applied to S2. In both scenarios, we consider brick as material to be ejected. Each simulation lasts 30 s. GridSort is compared with predictive tracking using Kalman filters with CV motion models for both prediction in MTT and to the nozzle array. For evaluation, we consider the true negative rate $\mathrm{TNR} = \mathrm{TN}/(\mathrm{FP}+\mathrm{TN})$ and the true positive rate $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, where positive particles are those that should not be ejected. A high TNR indicates a high purity of the non-ejected fraction, whereas a high TPR indicates a high purity of the ejected fraction.

The results are given in Tab. 2. GridSort achieves very similar TNRs and TPRs compared with predictive tracking, although not outperforming it. As expected, the accuracies of both approaches are lower on the high mass flow scenario S2 than on S1 (see Bauer et al. (2022) for a detailed discussion). Surprisingly, although never trained on S2, GridSort was still able to perform similarly to predictive tracking. In summary, with a TNR of 97.5 % and a TPR of 92.2 % for S1, GridSort can be considered as highly accurate.

[3] Evaluated on an NVIDIA GeForce RTX 2080 Ti.

Table 1. Pixelwise class accuracies of the proposed ConvLSTM network. BR, SL, and BG abbreviate brick, sand-lime brick, and background, respectively.

		(a) Real-world data			(b) DEM data		
		Predicted			Predicted		
		BR	SL	BG	BR	SL	BG
Actual	BR	0.96	4.8×10^{-5}	0.037	0.97	0.004	0.03
	SL	0.00026	0.89	0.11	0.0086	0.96	0.028
	BG	0.015	0.014	0.97	0.017	0.0068	0.98

Table 2. Comparison of sorting accuracies.

	predictive tracking		GridSort	
	S1	S2	S1	S2
TNR	0.982	0.956	0.975	0.953
TPR	0.937	0.751	0.922	0.695

7. CONCLUSION

We proposed GridSort, a new approach for optical sorting that determines nozzle activations by forecasting a segmented image of the scene at the nozzle array based on a sequence of area-scan images. For forecasting, we proposed a ConvLSTM-based neural network inspired by the grid predictor model of Schreiber et al. (2019). We demonstrated that the network can be trained in an unsupervised fashion using the concept of a virtual nozzle array, i.e., by shifting and shortening the input image so that the target image can be observed by the area-scan camera. This eliminates both the tedious and costly manual labeling of training data as well as a setup phase requiring expert knowledge as needed by existing algorithms. Our proposed method shows high image forecasting and sorting accuracies up to 97 % on high mass flow scenarios as evaluated with the help of a DEM–CFD simulation of an optical belt sorter. It is therefore a valuable alternative to state-of-the-art predictive tracking, as it is the first approach that also allows forecasting particle extents, is potentially able to predict non-linear particle motion behavior, and its run time is almost independent of the number of particles. Future work may focus on applying GridSort to a real sorter, for example, by using appropriate hardware to meet the real-time requirements and experimentally adjusting the deflection windows to account for systematic biases arising from, e.g., the model and the influence of fluid forces.

REFERENCES

Bauer, A. et al. (2022). Towards a Feed Material Adaptive Optical Belt Sorter: A Simulation Study Utilizing a DEM-CFD Approach. *Powder Technology*, 411, 117917.

Bauer, A. et al. (2023). Benchmarking a DEM-CFD Model of an Optical Belt Sorter by Experimental Comparison. *Chemie Ingenieur Technik*, 95(1-2), 256–265.

Bruce, R.C. et al. (2021). The Impact of Optical Berry Sorting on Red Wine Composition and Sensory Properties. *Foods*, 10(2), 402.

Chen, C. et al. (2006). Dynamic Environment Modeling with Gridmap: A Multiple-Object Tracking Application. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, 1–6.

Dequaire, J. et al. (2018). Deep Tracking in the Wild: End-to-End Tracking using Recurrent Neural Networks. *The International Journal of Robotics Research*, 37(4), 492–512.

Friedrich, K. (2022). Sensor-based and Robot Sorting Processes and their Role in Achieving European Recycling Goals - A Review. *Academic Journal of Polymer Science*, 5(4).

Itkina, M. et al. (2019). Dynamic Environment Prediction in Urban Scenes using Recurrent Representation Learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2052–2059.

Lange, B. et al. (2021). Attention Augmented ConvLSTM for Environment Prediction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1346–1353.

Lotter, W. et al. (2017). Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. In *5th International Conference on Learning Representations, (ICLR)*.

Maier, G. (2022). *Bildfolgenbasierte Gewinnung und Nutzung partikelindividueller Bewegungsinformation in der optischen Schüttgutsortierung*. Ph.D. thesis, Karlsruher Institut für Technologie (KIT).

Maier, G. et al. (2021). Experimental Evaluation of a Novel Sensor-Based Sorting Approach Featuring Predictive Real-Time Multiobject Tracking. *IEEE Transactions on Industrial Electronics*, 68(2), 1548–1559.

Mohajerin, N. and Rohani, M. (2019). Multi-Step Prediction of Occupancy Grid Maps With Recurrent Neural Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10592–10600.

Pfaff, F. (2019). *Multitarget Tracking Using Orientation Estimation for Optical Belt Sorting*. Ph.D. thesis, Karlsruher Institut für Technologie (KIT).

Pfaff, F. et al. (2015). TrackSort: Predictive Tracking for Sorting Uncooperative Bulk Materials. In *Proceedings of the 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2015)*.

Pieper, C. et al. (2018). Numerical Modelling of an Optical Belt Sorter Using a DEM–CFD Approach Coupled with Particle Tracking and Comparison with Experiments. *Powder Technology*, 340, 181–193.

Pollithy, D. et al. (2020). Estimating Uncertainties of Recurrent Neural Networks In Application to Multitarget Tracking. In *Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2020)*.

Robben, C. and Wotruba, H. (2019). Sensor-Based Ore Sorting Technology in Mining—Past, Present and Future. *Minerals*, 9(9), 523.

Schreiber, M. et al. (2019). Long-Term Occupancy Grid Prediction Using Recurrent Neural Networks. In *2019 International Conference on Robotics and Automation (ICRA)*, 9299–9305.

Thumm, J. et al. (2022). Mixture of Experts of Neural Networks and Kalman Filters for Optical Belt Sorting. *IEEE Transactions on Industrial Informatics*, 18(6), 3724–3733.