Experiments in Fluids (2024) 65:153
https://doi.org/10.1007/s00348-024-03884-z

RESEARCH ARTICLE

Peak-CNN: improved particle image localization using single-stage CNNs

Philipp Godbersen[1] · Daniel Schanz[1] · Andreas Schröder[1,2]

Received: 6 June 2024 / Revised: 21 August 2024 / Accepted: 9 September 2024 / Published online: 8 October 2024

Abstract

An important step in the application of Lagrangian particle tracking (LPT) or in general for image-based single particle identification techniques is the detection of particle image locations on the measurement images and their sub-pixel accurate position estimation. In case of volumetric measurements, this constitutes the first step in the process of recovering 3D particle positions, which is usually performed by triangulation procedures. For two-component 2D measurements, the particle localization results directly serve as input to the tracking algorithm. Depending on the quality of the image, the shape and size of the particle images and the amount of particle image overlap, it can be difficult to find all, or even only the majority, of the projected particle locations in a measurement image. Advanced strategies for 3D particle position reconstruction, such as iterative particle reconstruction (IPR), are designed to work with incomplete 2D particle detection abilities but even they can greatly benefit from a more complete detection as ambiguities and position errors are reduced. We introduce a convolutional neural network (CNN) based particle image detection scheme that significantly outperforms current conventional approaches, both on synthetic and experimental data, and enables particle image localization with a vastly higher completeness even at high image densities.

1 Introduction

In the last decades, Lagrangian particle tracking (LPT, Schröder and Schanz 2023,) has evolved to a technique capable of volumetrically following a multitude of tracer particles even in complex turbulent flows. Classical 3D particle tracking approaches operating with simple triangulation techniques to reconstruct 3D positions from projections on several cameras were typically restricted to low particle image densities $N_i < 0.01$ particles per pixel (ppp) (Maas et al. 1993), resulting in particle tracks and corresponding velocity vector data at relatively low spatial resolution. Nowadays, 3D particle positions are deduced using advanced iterative triangulation methods, such as iterative particle reconstruction (IPR) (IPR, Wieneke 2012; Jahn et al. 2021,) which allow increasing the usable range of single time step 3D reconstructions to $N_i > 0.1$ppp, depending on the image quality. Using real-world images at such high particle image densities, even these state-of-the-art reconstructions will still show ambiguities, which are mainly stemming from lines of sight (LOS) from peaks on different cameras coincidentally crossing on volume positions where no real particle was located during imaging. With increasing ppp values, these ambiguities become more and more likely and produce false or "ghost" particles (Elsinga et al. 2011; Wieneke 2012) within the reconstructed 3D particle distribution. For time-resolved recordings, predictor/corrector schemes can be used to effectively seize the temporal information contained in a time series of particle images, as demonstrated by the Shake-The-Box algorithm (STB, Schanz et al. 2016,), which massively reduces the evolving ghost particle problem at higher ppp values. However, the starting point for all triangulation-based reconstruction methods is a detection of particle images on several camera

✉ Philipp Godbersen
philipp.godbersen@dlr.de

Daniel Schanz
daniel.schanz@dlr.de

Andreas Schröder
andreas.schroeder@dlr.de

[1] German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Bunsenstrasse 10, 37073 Göttingen, Germany

[2] Institute of Transport Technology, Brandenburg Technical University, Siemens-von-Halske Ring 15a, 03046 Cottbus, Germany

projections and an enhancement of this starting point will instantly yield improved results. A STB-based prediction and correction scheme need to reach its converged state for full performance where almost all particles can be predicted from existing tracks. Improvement of the starting point would make this process faster and more precise and could allow for processing of higher particle densities, where current approaches do not provide a sufficiently accurate starting point for the tracking scheme to reach a converged state at all. Especially in high-speed flow cases in industrial aerodynamics, where two- or multi-pulse particle tracking is required, an improved sub-pixel accurate 2D particle image detector is very beneficial due to the limited possibilities of using temporal coherence to enhance the results.

In order to differentiate between the 3D location of a particle and its 2d position projected onto a camera image, we will refer to the latter as "peak" within this paper. This is based on the terminology used by Wieneke (2012) likely due to the reuse of approaches initially devised to find correlation peaks for particle image velocimetry. The method we are introducing does not necessarily look for maxima in the image intensity in order to determine 2D particle positions. Nevertheless, we still utilize the term "peak" as a succinct stand-in for the 2D position of a particle projected onto a camera image within that cameras coordinate system.

An overview of approaches used to detect such peaks is given by Dabiri and Pecora (2019), Ouellette et al. (2006), including weighted averaging (Maas et al. 1993) and fitting of (Gaussian) functions (Mann et al. 1999). Found sub-pixel accurate 2D positions on the cameras are then used to triangulate the 3D positions of the particles using a calibrated camera model.

The current state-of-the-art IPR uses multiple passes of peak detection, triangulation and optimization of positions and intensities to increase 3D particle reconstruction performance. Even if a peak detector misses many true particle images initially, the problem gets easier with iterations, as the already found particles are gradually removed from the (residual) images. The goal of this approach is that peaks that were initially missing due to overlapping particle images or otherwise hard to identify can be recovered on later iterations when the residual images are more and more empty. Nevertheless, this strategy can still benefit from an improved peak detection algorithm. If more peaks are correctly found from the beginning on—and with higher accuracy—the IPR processing benefits, as triangulation ambiguities are reduced and tighter bounds on triangulation search radii can be used. Both these factors facilitate applying IPR at higher seeding densities than currently used or resulting in 3D particle reconstructions with a lower fraction of ghost particles at given particle image densities.

2 Existing peak detection approaches

Figure 1 shows a typical example of a camera image from an experimental LPT investigation. In order get access to the underlying particle distribution, a fully time-resolved 3D reconstruction of the particle tracks (using multiple camera time series) was performed using STB and the tracked particles at the corresponding time step were back-projected into the camera images. The red crosses therefore are not the result of a peak detection process, but signify a "ground truth," evaluated with the help of several cameras and temporal information. Peak detection algorithms try to extract the sub-pixel accurate locations of the particle image peaks as faithfully to the real particle distribution as possible. The conventional peak detection algorithm currently used within the DLR implementation of STB, described by Jahn et al. (2021), calculates a cubic interpolation of the camera image and then identifies local intensity maxima within this interpolation via function fitting. Only intensities over a certain threshold are considered; this provides sub-pixel locations for potential particle peaks. We term this peak detector *CI* and primarily use it for comparisons with the introduced *Peak-CNN* method, based on a convolutional neural network.

Additionally, we have selected several peak detection methods from the literature in order to provide a fair evaluation and a comprehensive overview of available methods. Dabiri and Pecora (2019) provide a good overview of existing approaches, some of which we discuss further below. The OpenPTV project (Meller et al. 2019) provides a peak detector "target_rec," here termed *TR*. This detector appears to be of a similar class as *CI* albeit somewhat more sophisticated. Instead of just selecting an intensity threshold, additional parameters such as particle extent can be adjusted. As an additional variant for comparison, we turn toward an

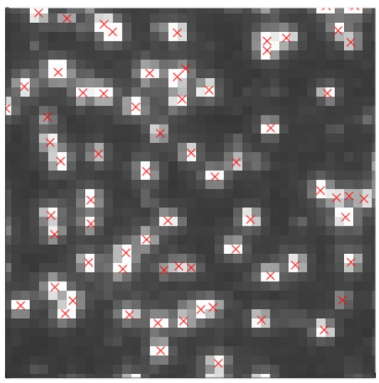


Fig. 1 Detail section of a LPT measurement image with particle peak positions obtained from back-projected tracked particle positions. The goal of a peak detector is to recover these positions from the measurement image alone, which can be a challenge due to overlapping particle images

image erosion based peak detector, described by Cardwell et al. (2011) (*IE*). The general working principle consists as a first step in a region detection scheme based on image erosion and dilation that identifies compact regions associated with particle image locations. In an ideal case, each region corresponds to one particle image and its pixels can then be used to fit a 2D Gaussian function in order to determine the peak location. For higher seeding densities, this is generally not the case as overlapping particle images form connected regions that often cannot be separated by the erosion approach any more. Boomsma and Troolin (2018) extend this with a multiple Gaussian fitting in order to handle multiple peaks within a region; however, this requires prior knowledge of their number. In their use case they directly integrate the peak detection into the tracking scheme and are able to predict these from extending prior tracks. As such it is not comparable to the pure peak detection approaches considered in this paper, but should be compared against a full LPT approach such as STB.

A very different approach to peak detection is taken by Cheminet et al. (2018) who attempt to invert the optical transfer function (OTF) of the measurement system using a non-negative least squares (NNLS) deconvolution based approach. If the OTF is sufficiently well known, this allows to recover the particles as perfect point sources, and by doing this on a finely subdivided grid, they achieve sub-pixel resolution. Ideally only one sub-pixel remains for each particle which can then be taken as the peak location. We found this approach to work exceptionally well on noise free synthetic data with known OTFs. Even very closely overlapping particles were still able to be resolved as separate peaks. With the addition of image noise and uncertain OTF information, the performance degraded. Unfortunately, this approach is computationally extremely demanding when compared to other peak detection schemes. A $N \times aN$ matrix with N as the number of pixels in the image and a as the subdivision factor is involved in the NNLS which quickly becomes infeasible to solve in a reasonable time frame for large image sizes. In our test, even a 32×32 px image sub-region with a subdivision factor of a=10 required about three minutes of calculation time. The authors discuss a modified approach to address the performance on larger images but were only able to reduce the effort by an average factor of 2.3 for densely seeded images. In contrast most of the other peak detection schemes discussed here are able to provide results for the full synthetic experiment with four cameras of 1300×1300 px each in less than a second. Due to these issues, we do not further consider this method within this paper despite its great potential in case massive computing resources can be made available.

The existing approaches presented so far were conventional whereas we introduce a machine learning-based method based on artificial neural networks. An early approach using a Kohonen neural network for peak detection was introduced by Carosone et al. (1995). Their approach consists of multiple stages where overlaps are detected and then shape estimation performed on candidates. They note the approach is fairly computationally intensive and unsuitable for smaller particles sizes. An application of fully connected neural networks in relation to peak detection is presented by Ouellette et al. (2006). Their network is trained to predict the 2D coordinates of a particle based on a 9 by 9 pixel image tile. The model is given an image patch centered on a conventionally detected particle candidate position and predicts a sub-pixel position relative to the tile. As such the model forms only part of a multistage process where candidate locations must first be identified and input tiles then generated. They compare this with other conventional refinement techniques and report favorable results for images with noise present. Since their study considered only seeding densities up to 0.01 particles per pixel (ppp), it is unclear how their method would perform in stronger overlap situations. A more recent approach is provided by König et al. (2020) who also present a method to obtain particle image positions using neural networks. Their circumstances are quite different to ours as they work with astigmatic PTV where particle image sizes are much larger. Particles are captured fairly large on the images as elliptic blobs since the goal is to recover their shape parameters in order to calculate depth information. The approach is a multi-step method where initial candidate locations are determined and then further analyzed. They are therefore able to utilize techniques from the object detection and classification community which are not suitable for the sometimes barely three pixel wide particles considered here. They also note that the approach struggles with overlapping particles, something we explicitly wish to be able to resolve. Sax et al. (2023) apply similar ideas for the use in defocussing PIV where again large particle image sizes are produced. Notably they break away from using neural networks in both of the stages and suggest a hybrid approach where one of the stages uses a conventional algorithm instead. Recently, another approach to peak detection has been described by Liang et al. (2022) who present a two-stage neural network approach to peak detection for PTV. Their approach with its two-stage nature is somewhat similar to prior techniques. In a first step, a candidate pixel is identified in the image. A small pixel region around this candidate is extracted and fed into a second model where this 7px by 7px patch is then used to calculate a sub-pixel coordinate. In some ways, this is similar to the earlier approach by Ouellette et al. (2006) or even the also previously mentioned work by König et al. (2020) but their novelty lies in the use of machine learning-based approaches in both stages of the detection scheme as well as a specific focus toward images used in PTV measurements.

In this work, we present our own CNN-based approach with a focus toward a large number of small and often overlapping individual particle images we typically face during densely seeded LPT experiments. As opposed to the multi-stage approaches described above, we achieve a single-stage solution that avoids handling issues involved in the training of multiple neural networks and the overhead of extracting image patches.

3 Convolutional neural network for peak detection

We utilize a CNN to perform peak detection on the camera images. At first glance, the peak detection problem is not a good fit for single-stage neural network approaches as the number of peaks per image varies. Directly outputting the found peak coordinates is therefore not possible as the number of output neurons would vary for each image. To solve this issue, we conceptually separate the problem into two different tasks. If we restrict ourselves to calculating only pixel accurate peak positions, we can treat the peak finding as a binary classification problem. For each pixel in the input image, we desire to know if it contains a particle peak or not. As the image size is known in advance and stays constant for all images in the measurement independent of the number of particles, this is quite suitable for a neural network approach. Such a binary classifier could then be used directly as a peak finder simply by assuming a peak in the center of each of the identified pixels.

By discretizing on the pixel grid, sub-pixel accuracy is lost so even if the classifier operates flawlessly individual peak position errors of up to $\sqrt{2}/2 \approx 0.71$ pixel are possible. Positioning errors of this magnitude are undesirable for triangulation purposes so that a strategy to recover sub-pixel accuracy is required. We do this by calculating sub-pixel offsets for each found particle image. These describe the position of the peak within the pixel relative to its center thereby restoring the ability to achieve sub-pixel accurate positioning. Our CNN model is able to complete both tasks at once in a single model using a combined cost function inspired by the YOLO ("You only look once," Redmon et al. 2016,) network architecture. This is a network designed to perform object detection, classification and bounding box computation all at once with a single pass. In its original form, it is targeted toward detecting a limited number of somewhat large objects such as cars, animals or household objects in an image. Together with the actual classification of detected objects, it has a very different target from what we are building toward, tens of thousands of tiny particles with strong overlaps, but can still provide design influence. The YOLO model similarly has the challenge of a variable number of objects per image that, once detected, individual attributes are to be predicted for. These attributes such as the objects class or its bounding box are only meaningful if there actually is an object at that location. They solve this by moving this discrimination into the cost function. The image is separated into a medium sized grid of sub-regions, and for each sub-region, the attributes are always predicted by the model, irrespective of classification status, and then simply ignored if there is no object predicted to be there. The cost function for training contains the classification loss for object detection over all candidate areas as normal, but any additional cost terms for object attributes are masked so that only the loss values for candidate areas with an actual object present contribute toward the total. This presents a powerful approach for combining detection tasks with attribute prediction for the variable number of detected objects within a single pass of a neural network. Our model follows these ideas but reduced down to a pixel individual scale instead of the coarse grid of candidate locations used in the original YOLO approach. The attributes predicted for each pixel are then not parameters describing a bounding box and the type of object but the sub-pixel offset of the peak within that pixel as shown in Fig. 2. This allows for a joint classification and regression task in each individual pixel built from the following expressions for the classification and regression loss values for a single pixel:

$$\begin{aligned} C_{ij} &= k_{ij} \log\left(\sigma(\kappa_{ij})\right) + \left(1 - k_{ij}\right) \log\left(1 - \sigma(\kappa_{ij})\right), \\ R_{ij} &= 1_{ij}\left(|a_{ij} - \alpha_{ij}| + |b_{ij} - \beta_{ij}|\right)/2, \\ \text{with sigmoid } \sigma(x) &= \frac{1}{1 + e^{-x}}, \end{aligned} \tag{1}$$

classification logit value κ_{ij}, true classification label k_{ij}, masking variable 1_{ij} which is one for a pixel containing a true peak and zero otherwise, predicted uv-sub-pixel offsets α_{ij} and β_{ij}, as well as the true offsets a_{ij} and b_{ij}, all for a pixel with index i and j. The masking variable 1_{ij} can be simply set to k_{ij} in our case as we perform a binary classification. The total loss is then calculated by

$$L = \frac{1}{N_x N_y} \sum_{i}^{Nx} \sum_{j}^{Ny} C_{ij} w_{ij} + \frac{1}{N_{\text{true}}} \sum_{i}^{Nx} \sum_{j}^{Ny} R_{ij}, \tag{2}$$

a mean loss over all pixels with N_x and N_y as image width and height in pixels and $N_{\text{true}} = \sum_i^{Nx} \sum_j^{Ny} k_{ij}$ as the number of pixels containing a peak. The weight w_{ij} allows to direct stronger attention during training to specific image regions. Such a weight can provide a "class weighting" commonly used for imbalanced datasets. In measurement images, there are generally less pixels that contain a particle peak than those without. For sparsely seeded images where empty pixels dominate, it can be necessary to put greater emphasis on

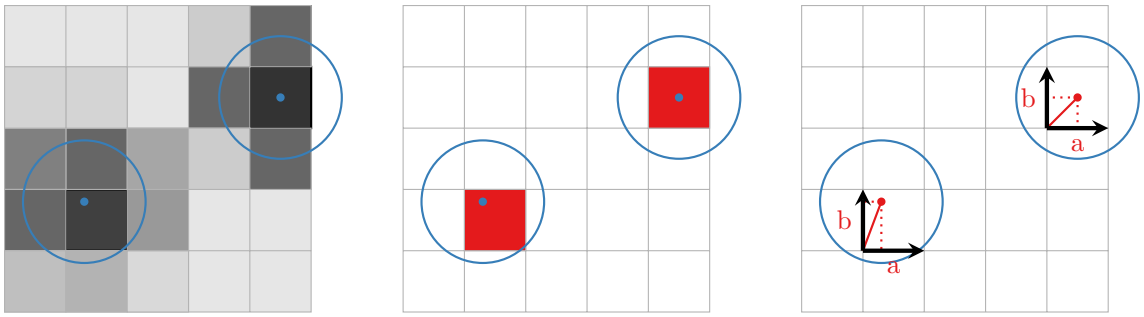


Fig. 2 Peak detection scheme with two idealized particle locations as blue circles. Starting from the actual discretized particle images (left), we can perform an initial peak detection by identifying those pixels which contain a particle peak (center). Sub-pixel capability can be regained by introducing an offset coordinate for each pixel that can address a peak contained within it relative to the lower left corner (right). This forms the internal representation of peaks for the CNN

those pixels that do contain particles to improve the training of the model:

$$w_{ij} = \frac{1}{2} N_x N_y / \sum_{n}^{N} x \sum_{m}^{N} y \begin{cases} k_{mn}, & \text{if } k_{ij} = 1 \\ 1 - k_{mn}, & \text{if } k_{ij} = 0 \end{cases} \tag{3}$$

We found it advantageous to not just increase the weight of the individual peak containing pixels alone, but also their immediate surroundings. Simply by convolving the weight mask with a 5 by 5 px Gaussian kernel, we can create a weight that focuses the training not just on the immediate peak but also provides elevated weight around it in a region where incorrect classifications are most critical for later triangulation.

One disadvantage of this approach is that no more than one peak can be addressed per pixel since only one offset coordinate pair is available. For extremely densely seeded flows where such events are not insignificant in number one could extend the above description from a binary classification approach to a multiclass problem where the classes correspond to $0, 1, \ldots, N$ particle peaks within the pixel for some predetermined N. The masking function 1_{ij} would then mask out N offset coordinate pairs as appropriate for the given class, e.g., 2 coordinate pairs for class 2. Due to the rarity of pixels containing multiple peaks such an approach would represent a highly imbalanced multiclass classification problem posing a challenge during training with special consideration required toward the weight function. For this reason, we currently only perform binary classification in our approach even if the architecture of the model itself could easily be extended to be able to express multiple peaks.

Input to the model is the camera image itself for which the model produces an output of the same height and width with three values per pixel: the binary classification and the sub-pixel offset in u and v. We can then collect the pixel coordinates of the positively classified pixels, modify them by their predicted sub-pixel offset and thus get a list of the uv-coordinates of all detected particles in the image. In addition to the camera image, we also provide data from the optical transfer function (OTF) as an additional input. This currently is implemented as the OTF evaluated in the center of the measurement volume giving four values describing the shape of the 2D Gaussian. These data are then resized to match the required size and concatenated to the data at that stage. To use a constant OTF for the entire image is a simplification as generally a calibrated OTF varies throughout the measurement volume. The model architecture we use would directly allow a OTF that varies over the camera image but this is not trivial to implement as one cannot simply project a volumetrically calibrated OTF (Schanz et al. 2012) onto the 2D camera image, as the OTF typically varies along the lines of sight. For this reason, we currently utilize just the single value evaluated in the center but the approach would allow for pixel individual values from camera calibration or other spatially varying values such as 2D disparity maps to be included into the model. Availability of the OTF is not a hard requirement for the proposed method. If no such data are available one can omit this input to the model and still obtain reasonably good results.

The general approach can be implemented as a simple series of convolutional blocks utilizing batch normalization (Ioffe and Szegedy 2015) and ReLU activation functions (Glorot et al. 2011). Padded 2D convolutions are used resulting in an output with the same height and width as the input image. While such a simple setup already showed promising results, we use an extended architecture within this paper as shown in Fig. 3. This is based on the U-Net architecture (Ronneberger et al. 2015) originally designed to provide segmentation of biomedical images. Instead of just performing straight convolutional connection where activation maps stay the same size as the input image, the U-Net adds a downscaling path which consists of successive convolutional and pooling layers. Whereas the activation maps start out with a

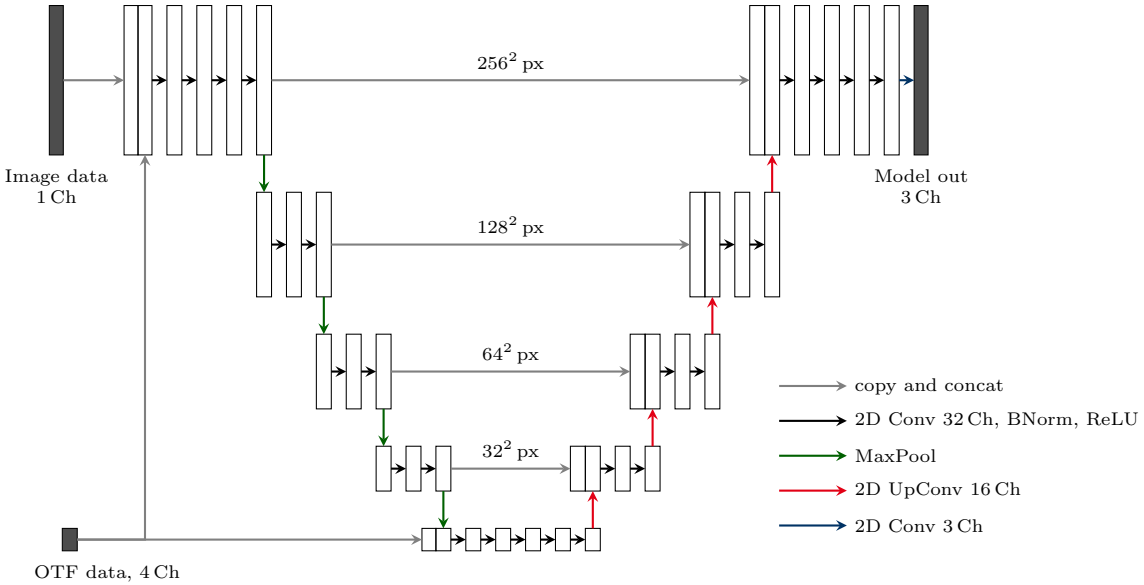


Fig. 3 Model architecture for our machine learning-based peak detector in form of an U-Net. Inputs are the image data as a single channel gray-scale image and the OTF data currently in form of a set of four numbers parameterizing a 2D Gaussian. The convolutions have a kernel size of 3 by 3 and generally use an output channel size of 32, reduced to 16 for the penultimate and 3 for the final output

size corresponding to the input width and height in pixels, an activation map at the end of the downscaling path is reduced to a size of 16 by 16. This allows the model to condense down information from a more global level such as an average value over the entire image, or larger regions such as differences between the left and right side. The condensed activation map is then fed through an upscaling path to reach the final image output size again. The architecture employs cross-connections between the two branches of the U-Net by concatenation of activation maps from before each downscaling step to the corresponding result after each upscaling step. These are essential for our application as they preserve fine scale information through the model which is critical for the tiny size of the particle images. In order to preserve GPU memory and to prepare for future applications involving variable input image sizes, a tiling strategy is used where inputs to the model are always 256 px by 256 px image tiles. For many of the test cases, this model architecture is probably larger than necessary and could benefit from further fine tuning of the amount of layers and channel sizes but we wanted to provide a consistent architecture for all cases. The current setup can serve as a starting point for problem specific optimizations.

The actual peak detection process via a trained model can then be applied to an image by taking the first channel of the model output which after application of a sigmoid contains the binary classification probability on the pixel level. Here a suitable threshold needs to be selected at which a pixel is considered to be activated. Once a pixel is determined to contain a particle peak, we can utilize the sub-pixel offset values from the two other channels of the model output to calculate the final peak position as an offset from the pixel center position. The classifier threshold value between zero and one is the main parameter to adjust the peak detection scheme once the model has been trained and essentially represents how certain we require the model to be when claiming the existence of a peak at a particular location. By varying this parameter, trade-offs in the sensitivity and specificity of the model can be explored. Beyond the adjustment of the peak detector, this availability of a certainty value for each peak could provide useful information for downstream components in an LPT processing scheme such as allowing the shaking procedure of an STB processing to provide less weight to the camera where the peak detection is less certain.

Only a limited number of measurement images is needed to build training data since each image with its large amount of pixels provides a multitude of information for training as all pixels are, except their immediate neighborhood, mostly independent from each other. This is not an issue when using synthetic data as essentially unlimited data can be generated but can be important for real-world data if only a short sequence is available for training. The generation of the training data for experimental cases is laid out in the next chapter. In some tests just ten images provide enough data to train a satisfactory model for an experiment. Based on the amount of available data, none of the data augmentation methods commonly used for images to increase the amount of available training instances, such as adding rotated or mirrored images, was used.

4 Evaluation on test cases

We evaluate the performance of the suggested CNN detector on several test cases comprised of synthetic and real-world experiments. Synthetic experiments have the advantage of full control over imaging parameters and availability of actual ground truth data to evaluate results as well as for use in training. However, synthetic images typically remain an imperfect imitation of real-world images as not all complexities of the imaging process are fully modeled. While this is a general issue affecting the comparison of LPT methods and their components, this is especially relevant to the method utilizing supervised learning we are proposing. Ground truth data are not just required when evaluating the performance of the method but also for training the model itself.

As such we wish to present results evaluated and trained on actual real-world experiments. The main challenge for this approach is the generation of ground truth data for such real-world images. Since the absolute ground truth data are not available, we utilize a pseudo ground truth based on a time-resolved STB evaluation of the experiment as described by Novara et al. (2023) instead. If all particles are successfully tracked by STB then we can simply select an individual time step and back-project the 3D positions of the particles onto the camera image, thereby generating particle peak positions. This approach relies on utilizing the additional information gained from the temporal relationships, the multi-camera views on a single particle and the performance of the STB algorithm in order to provide near perfect results from the view of individual images. The key idea is that a well-converged STB run barely relies on the performance of its peak detector anymore as almost all particles are positioned by predictions and corrections from existing tracks. As such it is capable of resolving particle overlap situations that would be impossible to distinguish on individual images, simply due to the fact that these particles belong to tracks that stay well separated before and after. This is of course an unfair advantage when viewed from the standpoint of methods restricted to individual images but it is exactly this property which makes such data so well suited for judging and training the single-image-based methods. Such results are not perfect however, and one must keep in mind that only a pseudo ground truth is produced. For the training of models, we do not expect that the small differences between actual ground truth and pseudo ground truth will have much effect since cost functions are calculated as a mean over all pixels. We use the same approach to generate the data used to judge the different peak detection algorithms and errors in this pseudo ground truth will slightly affect these results as well. However, such effects should be small and will affect all algorithms equally, so comparability is preserved.

In the following all ground truth data for training and evaluation of real-world particle imaging is obtained this way unless otherwise noted. This works well for benchmarking the method on existing datasets, but ultimately we wish to apply this method to optimize running LPT experiments or start processing freshly generated datasets. After all, how could an LPT evaluation gain from a higher quality peak detection, if the underlying method requires an accurate tracking solution of the same data as training data? These concerns of putting the presented approach into practical application will be addressed in Sect. 5. For now we will focus on evaluating the performance of the approach and assume the availability of (pseudo) ground truth data for training. Training data can be generated from consecutive images with little danger of producing similar inputs due to the complex movement of the particles. Particles move a distance of several pixels between images, and together with their turbulent movement and the projection onto the 2D image plane, this results in strongly different peak positions and overlaps situations from one image to the next. For experiments involving very laminar flows or little temporal separation between images, one could select images at a greater temporal spacing or even random sampling if necessary. The actual training label input for the model is generated from the labeled training images using the reverse procedure used to obtain peak coordinates from model output. A list of peak coordinates for a given image is transformed into the internal *Peak-CNN* representation shown in Fig. 2 by rasterizing the peak coordinates onto the pixel grid and then computing their sub-pixel offsets within the respective pixel. One obtains three-channel data of the same height and width of the image data this way where the first channel contains a value of one if there is a peak in that pixel and zero otherwise. For those pixels where there is a peak, the values of channel two and three are set to its sub-pixel offset values or left zero otherwise. An additional training data input is the OTF data which is simply a set of four numbers for each camera obtained during calibration of the experiment. Models defined by the architecture described in Fig. 3 are then individually trained for each test case using the ADAM optimizer (Kingma and Ba 2017) for 40 epochs corresponding to about an hour of training time on a consumer grade RTX 2080 GPU. The amount of training data and thus the needed time varies depending on the number of cameras and the size of their images. Especially synthetic cases with only four cameras and smaller images than the real-world cases can be trained quicker. However, we observed no tendency of over-fitting the model.

Judging the performance of peak detection approaches requires some discussion of the metrics used. An optimal peak detection finds all particle peaks within an image without giving any erroneous peak locations. Metrics of how to discern between true and false particles have

been utilized in the realm of benchmarking 3D particle reconstructions and the same approaches can be used here too, by simply reducing the number of dimension to two (the image plane). In the following we will refer to peaks/particles in the ground truth as *true* peaks/particles and the peaks/particles found on the image to be judged as *detected* peaks/particles. For legibility only particles are mentioned in the rest of this section.

Based on the returned result of a particle reconstruction and known ground truth values, we can determine three classes of particles. Missed particles (false negatives, FN) are those true particles where no detected particle is located within an appropriate search radius around the true particle. Found particles (true positives, TP) are then those true particles where a successful match is made with a detected particle. This number should always complement the number of missed particles. False particles (false positives, FP) are those detected particles where no match can be made to a true particle. Some care must be taken when determining the number of found and false particles in order to match each true particle at most once. A situation where two particles are detected in the vicinity of a single true particle, only the one closest to the true particle should be considered as the correct one whereas the second one should count as a false particle even if located within the search radius. However, in case a second true particle is located within the search radius of the second particle, it will be considered a found particle. We use the same implementation as Novara et al. (2023) of the rules used for the first LPT challenge (Sciacchitano et al. 2021). Whereas the CNN community typically uses quantities such as precision and recall in the scoring of classifiers, the predominant error value used in the IPR and LPT literature mentioned above deal in percentages with respect to the ground truth number of particles. To be more specific, with the number of true particles P and the number of detected particles D = TP + FP as sum of true positives and false positives we can evaluate the detection performance by the factors:

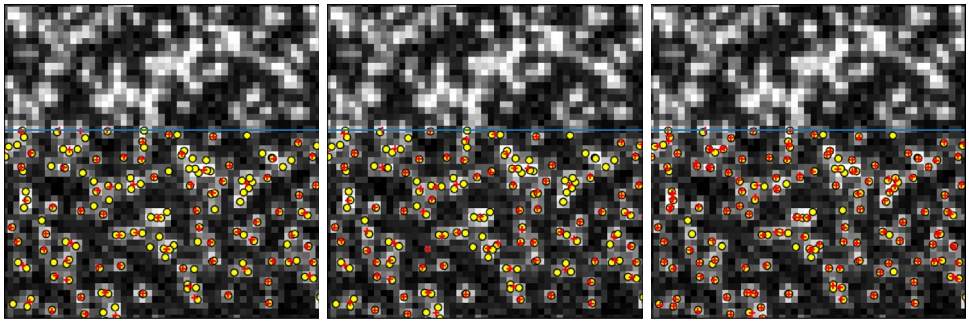
- Positive predictive value or precision as TP/D = TP/(TP+FP)
- True positive rate, sensitivity or recall as TP/P. or
- Found particle rate as TP/P (same as recall)
- False (Ghost) particle rate as FP/P,

with the last two using terminology from the IPR development (Wieneke 2012; Jahn et al. 2021). Note that while the found particle rate is the same as recall, the ghost particle rate is calculated by dividing by P instead of D. This distinction is a minor one, essentially just whether to normalize the number of false particles by the number of detected particles or the number of true particles. Within this paper, we will adopt this convention and utilize the ghost particle rate for evaluation of particle and peak reconstruction performance unless otherwise noted.

4.1 Synthetic test case

To evaluate our approach, we first utilize a synthetic experiment based on the one used by Jahn et al. (2021) which in turn is based on the synthetic setup in the original IPR paper by Wieneke (2012). Within this paper, we utilize their Case II which is a variant with moderate image noise and variations in the particle intensities. For this synthetic experiment, we generate separate test cases at seeding densities in the range of 0.05 to 0.15 ppp with 30 snapshots for each. Snapshot generation is driven by the seed value of the random number generator used in the synthetic experiment. Each synthetic measurement snapshot corresponds to a certain seed value. By selecting some number to define our evaluation measurement and then only ever incrementing the seed during training data generation, we ensure that the evaluation data have not been seen by the model during training. We combine the data of the different seeding densities and train a combined model to also show the ability to generalize to a range of different seeding densities. But this is not necessary and for this synthetic case models trained on just one seeding density also gave similar results as the joint model when applied to other seeding densities. Evaluations shown here are for the joint model. For a broad comparison, we also apply the three standard approaches introduced in Sect. 2 (our cubic interpolation-based *CI*, OpenPTVs "target_rec" *TR* and the image erosion based *IE*). A visualization of the results of peak detection is shown in Fig. 4 for the different detectors on the 0.1 ppp case. The overlapping particle images form connected blobs so that the conventional peak detectors are only able to separate them in some cases. They often place a single detected peak in the center of such a blob. The CNN-based detector is much more successful in resolving such overlap situations resulting in a greater amount of peaks that are correctly detected.

A quantitative evaluation is shown in Fig. 5 for three different seeding densities in terms of found peak rate and false peak rate. For each detector, a primary parameter is varied that modifies its aggressiveness. In the case of the conventional detectors, this is an intensity threshold given in counts. For the *Peak-CNN* model, the classifier threshold is used. The conventional detectors show very similar behavior. Decreasing the intensity threshold allows more particles to be found up to a point where further decrease only leads to a sharp increase in false peaks. The higher the seeding density the more peaks disappear behind this cliff edge. This is due to the limited ability of these detectors to resolve strong particle overlap situations which they register as a single peak as shown in Fig. 4. Decreasing the intensity threshold does not improve their ability to separate the overlap so the



(a) *CI* (b) *TR* (c) *Peak-CNN*

Fig. 4 Results of different peak detection algorithms for the synthetic case at 0.1 ppp. Shown is an image cutout with ground truth positions in yellow and detected positions in red. No markers are drawn in the upper third. At this seeding density particles strongly overlap. Approaches **a** and **b** struggle with closely overlapping clusters whereas our *Peak-CNN* approach **c** is much more successful in discerning the peaks in such scenarios

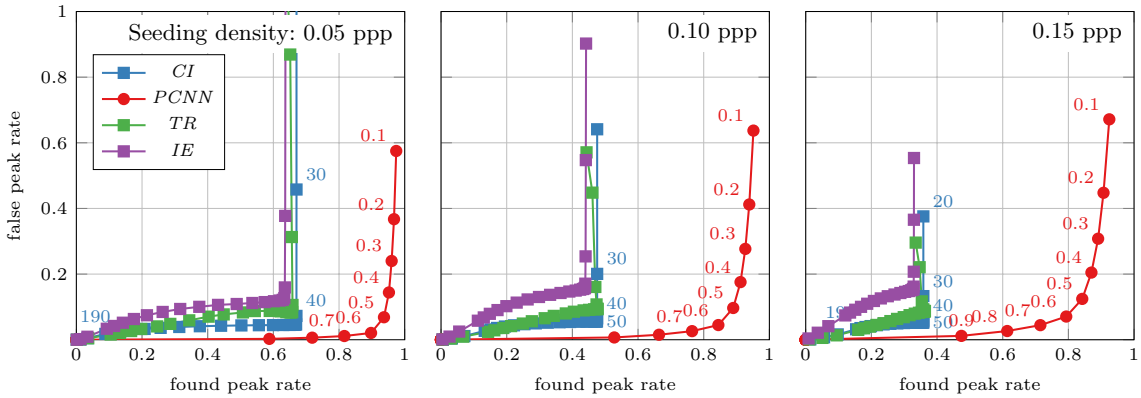


Fig. 5 Performance of various peak detectors on the synthetic case at varying particle image densities. Annotations show the selected parameter for each point: The intensity threshold for the conventional peak detectors (squares) and the classification threshold for the *Peak-CNN* model (red circles). The same CNN model was used regardless of seeding density

additionally reported peaks are mostly false peaks found in noise. Our *Peak-CNN* detector on the other hand is able to recover a majority of the true peaks even at high seeding densities. It too registers more false peaks as the classifier threshold is decreased but this process is more gradual and is accompanied by an increase in found particles, allowing to select trade-offs between precision and recall. The *Peak-CNN* detector is much less affected by the increase in seeding density and can still recover more than 80% of peaks at 10% false detections in the 0.15 ppp case. We also examine the positioning error of correctly found peaks for each method. As shown in Fig. 6 the CNN-based approach also displays a lower peak positioning error than the other investigated methods. Not only more peaks are found correctly, their positioning is also improved. These improvements come at a moderate increase in evaluation time as shown in Table 1. Our *Peak-CNN* approach takes almost twice as long to evaluate as the simpler *CI* approach but still well below one second for this individual four camera snapshot. The CNN-based detector is running on the same GPU used during training whereas the others are executed on the CPU (AMD Ryzen 9 3950X). Times for the *Peak-CNN* include the transfer of the data to the GPU and back to provide a fair comparison. It can be run on the CPU instead as well but with significantly increased evaluation times. If this is necessary one should attempt to reduce the complexity of the model architecture as much as feasible in order to obtain acceptable evaluation times. Considering that many more peaks are found by the CNN detector we additionally report the time per found peak which gives similar values. The model architecture we use has not been optimized for evaluation speed so if detection time is critical this likely can be improved upon by use of a more reduced model. The fairly large evaluation times of the two other detectors are most likely due to additional overhead introduced by our implementation and not necessarily representative of the respective methods. It was necessary to apply an image tiling strategy for the *TR* detector from OPTV for example as

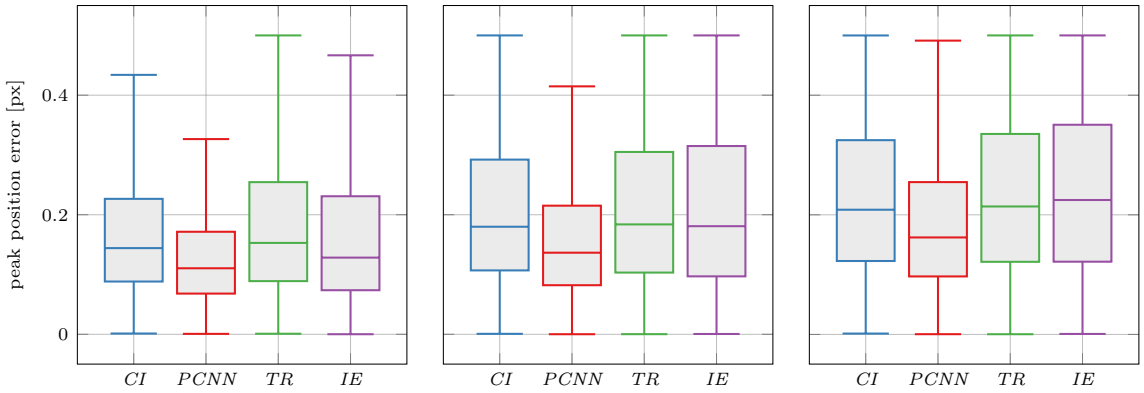


Fig. 6 Peak positioning errors of correctly found particles for the synthetic case at seeding densities of 0.05, 0.10 and 0.15 ppp (from top to bottom). For each detector, a suitable parameter was selected from Fig. 5

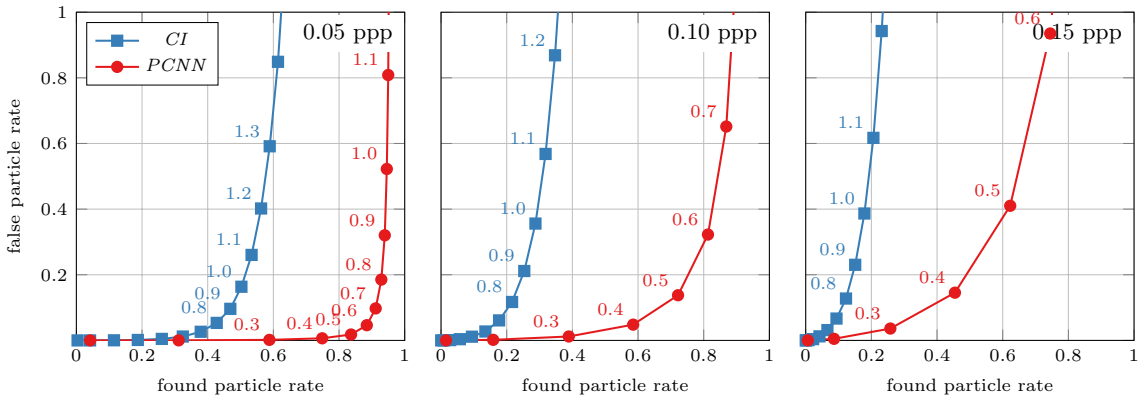


Fig. 7 Improved peak detection also translates to better triangulation performance. For each detector, the best parametrization according to Fig. 5 is selected and the resulting peaks are used for a triangulation procedure. Varied along the curve is the triangulation radius in pixel

Table 1 Run-times for the different peak detection methods on the synthetic 0.1 ppp case. For the CNN model, this includes the data transfer to the GPU and back

Method	Time [s]	Time/found peak [μs]
CI	0.30	1.56
Peak-CNN	0.58	1.62
TR	5.16	26.82
IE	78.49	443.63

the program is restricted to a limited number of found peaks. In terms of peak detection, the three conventional detectors show very similar results and we do not intend to make any claims as to their relative performance among each other. Some have additional parameters which can be individually fine-tuned for a given case affecting their ranking but no significant change in their overall performance was observed by us during some experimentation. Our intent here is to show that the conventional peak detector *CI* also used by Jahn et al. (2021) gives comparable performance to other peak detectors of its class from literature and can serve as a fair comparison for the rest of the paper while we will omit the others for clarity.

Beyond examining the identified peaks on their own, we are also interested in the results of a triangulation using the peaks from the multiple cameras of a setup. We utilize the same triangulation procedure as used by Jahn et al. (2021) in the following, but no position optimization (shaking) or iterative processing is applied in order to isolate the effect of peak detection on triangulation. This quantifies the improvements when applying *Peak-CNN* to standard PTV approaches which are not using IPR. In order to investigate the triangulation performance, we fix the detection parameter varied in Fig. 5 to a suitable value for each method and now vary the permissible triangulation error. The results are shown in Fig. 7. The gains in 2D peak prediction performance translate into improved triangulation results; the relative improvement between the methods is even greater

here. This can be understood, as a successful triangulation of a particle requires its presence in the cloud of identified peaks on all N used cameras of the system. The likelihood to meet this condition scales with the found rate to the power of N. The triangulation actually performs better than this relation, which is likely attributed to single peaks in overlap situations being used to successfully triangulate multiple of the involved particles. In case of *Peak-CNN*, the greater number and better positioned peaks as an input to the triangulation significantly improve the resulting 3D particle reconstructions. Much smaller permitted triangulation errors can be selected, which reduces the ghost rate. Using this approach, it was possible to correctly triangulate more than 70% of particles for the 0.1 ppp case with a ghost rate below 20%. Even for the 0.15 ppp case about 50% of particles were still correctly triangulated at a similar ghost rate. The *CI* approach yields 25% and 15% for the two cases, respectively.

4.2 RBC test case

The second test case and first real-world measurement is data from a Rayleigh–Bénard convection (RBC) experiment by Bosbach et al. (2021), obtained in a cylindrical volume with height and diameter of H = D = 1.1 m (Fig. 8). The helium filled soap bubbles (HFSB) used as tracers together with an array of LEDs for illumination result in high quality particle images, leaving particle overlap as the main challenge for peak detection in this experiment, especially at higher particle image densities. This experiment is particularly well suited for the evaluation of peak detection methods due to the closed measurement volume. Particles stay in the measurement domain for their entire lifetime, allowing for very long tracks to be obtained to serve as ground truth for the peak detection performance. This reduces the reliance on peak detection in the ground truth generation even further as no additional particles are entering the domain as would be the case in a wind tunnel experiment. The very long measurement times together with a finite lifetime of the soap bubble tracers inherently introduce a systematic change in particle image density. By selecting images from different times during the measurement, we can obtain test cases at different ppp values with an otherwise identical imaging setup. Particle image density is not constant across an individual image due to the cylindrical measurement volume leading to a decrease of imaged particles toward the edges. To be able to do comparisons at a more even particle image density distribution, we select a 300 px by 300 px region in the center of the image and only compute peak detection performance there.

As shown in Fig. 9 the 30,000 available snapshots for this experiment allow to examine a large variation from 0.01−0.18 ppp. Detectors are still applied to the full camera images, only the scoring is done in this central region where particle image density is highest and most even. For scoring the performance after triangulation, this restriction to the center is not as meaningful as the 3D particle positions are evenly distributed within the cylindrical volume so that all particles are used.

With this correspondence of measurement time step to seeding density, we can define seeding density variant cases by selecting data around time steps 100, 5000, 10000 and 15000 corresponding to seeding densities of 0.18, 0.12, 0.07

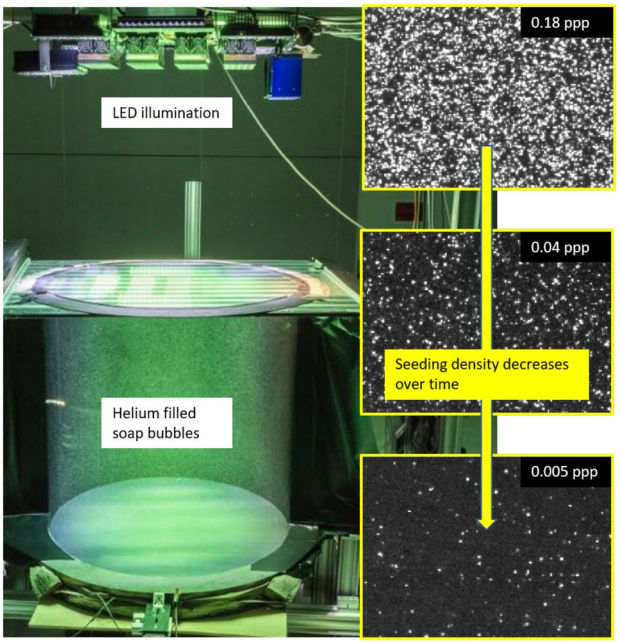


Fig. 8 Picture of the Rayleigh–Bénard convection experiment used for the RBC benchmark test case. The closed cell together with the finite lifetime of the bubbles provides a large range for the particle image densities over the long measurement duration (see column of particle images (right))

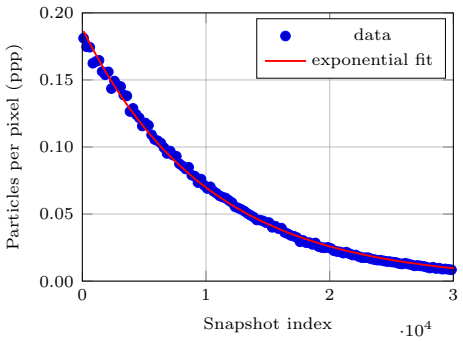


Fig. 9 Particle image density for the RBC dataset evaluated in a 300 px by 300 px region in the center of the image plotted over the experiment time steps. The density decreases due to the limited lifetime of the soap bubbles. Values were determined using back-projected tracked particles

and 0.04 ppp. For each variant 30 consecutive time steps are used as training dataset with the 5 following for validation. When comparing the performance of the different peak detection methods, we use data from a time 10 steps ahead of the starting point of each training data in order to ensure the use of images not seen by the model during training.

The results of evaluating detector performance are shown in Fig. 10. Over the entire range of particle image or seeding densities studied, the CNN peak detector displays superior performance. The conventional peak detector again only finds a significantly smaller amount of particles. Even by lowering the detection threshold, it is not possible to identify more particles. Interestingly, the large increase of ghost particles seen in the synthetic test case is not present here. Especially for the high seeding density case little variation over the intensity threshold is observed for the conventional approach. This is due to the fairly low noise level of this measurement together with the large amount of overlap of the particle images. Detection results for both methods are slightly worse than those for the synthetic data in Fig. 5, as expected.

The *Peak-CNN* detector covers a larger range of values allowing to select desired performance characteristics. Even in the very densely seeded 0.18 ppp case about 70% of peaks are correctly found for a false peak rate of 20%. The conventional detector is unable to reach 40% of found peaks here not due to an increase of false peaks but due to its inability to resolve the strong overlap situations. Just as for the synthetic case the improvement in peak detection performance also translates to improved results for 3D particles after triangulation as shown in Fig. 11. At the highest seeding density, *CI* is only able to reconstruct approx. 15 % of the true particles while accepting a ghost rate of 50 %. The *Peak-CNN* yields 50 % of correctly triangulated particles in the same scenario.

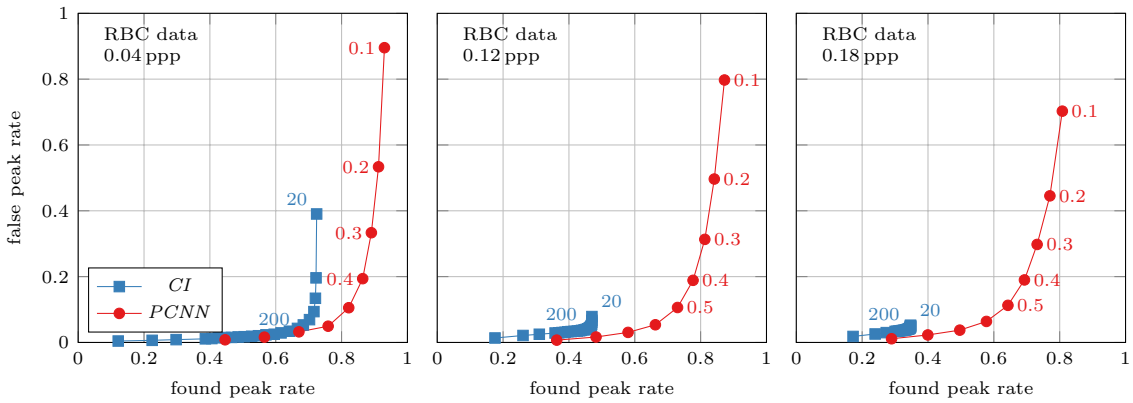


Fig. 10 Peak detection performance on a central region of the RBC test case at various seeding densities. Annotation indicates the variations of the detection threshold along the curves

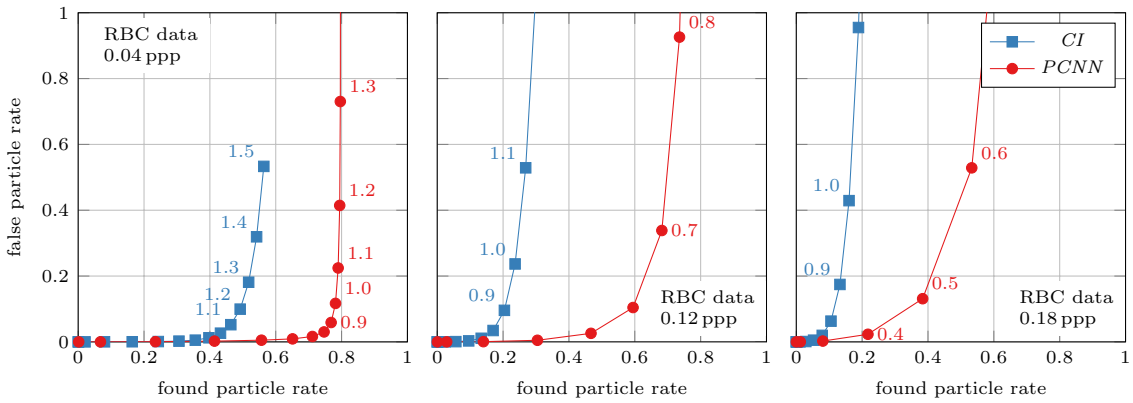


Fig. 11 Triangulation performance on the RBC test case at various ppp values. Annotations show triangulation radius in pixels. Evaluation is now done over the whole volume but the ppp label is kept the same to preserve comparability with Fig. 10

4.3 Jet dataset

The RBC experiment demonstrates the abilities of the proposed method at very good particle imaging conditions due to the close to ideal reflection behavior of LED illuminated HFSBs. However, PIV and LPT experiments are often conducted under much harsher imaging conditions, often using µm-sized oil or Di-Ethyl-Hexyl-Sebacat (DEHS) droplets as tracers, illuminated by a pulsed high-power laser. Interference effects from the coherent light source result in more complex particle images in overlapping situations while the Mie scattering off the droplets results in greater variation in particle image intensity across the various camera viewing angles. We have selected a multi-pulse STB experiment of a high subsonic velocity jet in air (Manovski et al. 2021) as a representative for problematic imaging conditions (Fig. 12). This experiment was conducted with small DEHS droplets illuminated by linearly polarized laser light for separation of multi-pulse images, adding to imaging quality challenges. Section 5 contains an explanation on how the ground truth data for such a multi-pulse experiment were acquired. Here we will again assume its existence for now. As we used volume self-calibration (VSC) images at low-seeding density (0.005 ppp), the dominating source for errors in this case is the image quality, rather than overlap situations. As shown in Fig. 13 the CI detector works reasonably well in this very sparsely seeded test case. Still, it is only able to identify approx. 71 percent of the correct peaks. The CNN-based detector yields an around 15 percent higher found rate at a given ghost rate. An even higher effect can be seen in the average position accuracy, which is increased from 0.25 px for *CI* to 0.11 px for *Peak-CNN*. This value is comparable to the one gained from the noisy synthetic data at low particle image density and will enable a significantly more robust 3D reconstruction, especially combined with the higher found rate.

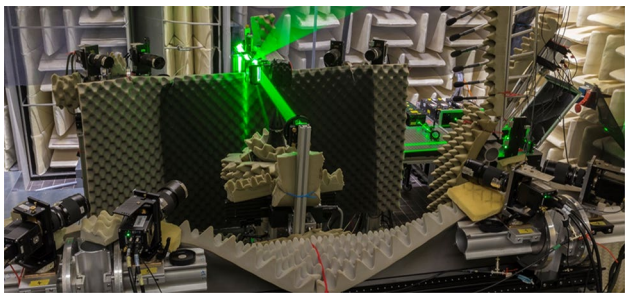


Fig. 12 Experimental setup of the subsonic jet case

5 Training data generation for real experiments

So far we have demonstrated the performance of our proposed method on existing experimental data. If the approach is to be used in the actual processing of new experiments, the question of sourcing of training data arises.

From the results above, we see that the approach of generating ground truth data via LPT evaluation works well. However, in order to generate such an evaluation one has to rely on a conventional peak detector. If the gained *Peak-CNN* detector would be limited to the imaging conditions it has been trained on, the improvements to the current status quo would be incremental. Therefore, we explore several more involved strategies for creating training data that exceed the conditions that are solvable using conventional peak detection algorithms. Two of these are based on using tracked particle data for label generation while the third one is an alternative approach that does not require tracking input data, but is using the *CI* peak detector for training data generation. In the remainder of the manuscript, models trained by tracked data are labeled with the subscript "tracks," while ones trained by simple peak detection are labeled "peaks."

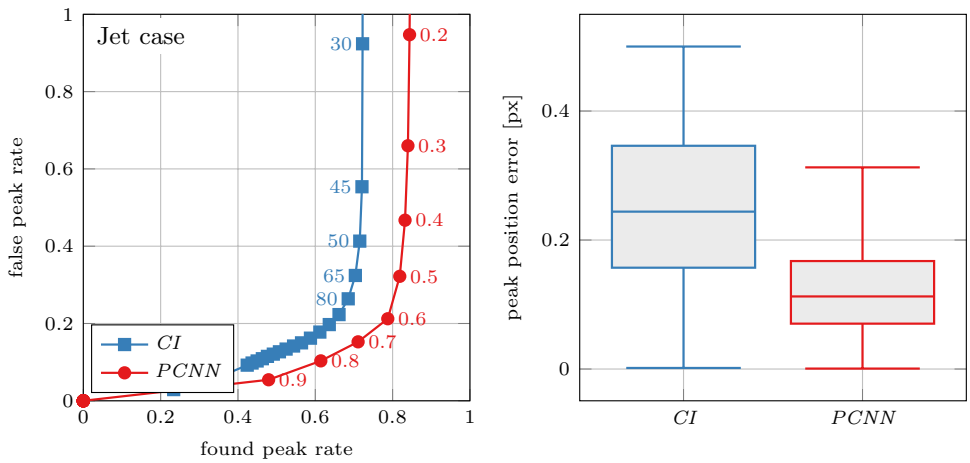


Fig. 13 Peak detection performance (left) and peak position accuracy for the jet case of our *Peak-CNN* and the standard *CI* peak detectors

5.1 Sourcing training data from tracking evaluations

The first approach relies on the assumption that the trained model can be extrapolated to higher seeded conditions to a certain extend. The concept is to train a model on a moderately seeded case and simply apply it to conditions with higher seeding density. For the RBC experiment, this approach was tested by applying the model trained by tracks at 0.04 ppp ($Peak-CNN_{tracks@0.04}$) to images at 0.12 ppp. Figure 14 shows the triangulation performance of this attempt (green curve) in comparison with the 0.12 ppp model ($Peak-CNN_{tracks@0.12}$, red curve) for pure peak detection, as well as triangulation. The performance of both models is very close. Presumably the training data at the lower seeding density contain enough particle overlap situations that the model is able to successfully separate them even at the higher density. Even a model trained at 0.01 ppp ($Peak-CNN_{tracks@0.01}$, purple curve) shows only a slight degeneration of reconstruction performance.

A modified version of this approach is to artificially augment the training data to more closely match the images to the conditions during the actual experiment. Using images at a (very) low-seeding density ensures a reliable tracking, even with basic peak detection and tracking schemes. In a second step, several of the images can be summed up (stacked) to produce images at a higher seeding density. The ground truth from the tracking process is generated by stacking the back-projections of the particle positions from the same time steps used for the image stacking. The stacked time steps can be completely unrelated in time, as only the particle positions and the related image peaks are relevant for the process. This approach was again tested on the RBC data by using a dataset at 0.01 ppp and stacking both the images and the tracking result twelve times, leading to an effective seeding density of 0.12 ppp. As before, the gained model ($Peak-CNN_{tracks@12*0.01}$) was applied to 0.12 ppp image data. The results are plotted as the orange curves in Fig. 14. It can be seen that the performance is for most values of the detection parameter a bit better than the $Peak-CNN_{tracks@0.01}$ model. When allowing many false peaks, the results are comparable to the $Peak-CNN_{tracks@0.12}$ model. For triangulation the stacked model performs a bit worse, as we chose a relatively low detection threshold, where the stacked model performs slightly worse. As the gains of stacking tracked data are not pronounced, we do not pursue the approach further; however, the concept of stacking images is picked up again in the next section.

The results from the different approaches using tracked data document the stability of this approach. The tracking process ensures that overlapped particle situations are resolved and can be learned by the model, even using low-seeding density images. During the setup of an experiment, it needs to be considered that low-seeding density images are required for the calibration of the model. Typically, such images are recorded anyhow in order to perform volume self-calibration and to calibrate the OTF. These images can be reused for the training process.

A special treatment is required for multi- or two-pulse STB measurements (Novara et al. 2019, 2023), where only very short temporal segments are captured. In these cases normally no converged tracking system can be established due to the limited number of available time steps. For training data generation, one would either have to create a particular measurement run with time resolution (e.g., under wind-off conditions, so that the particle shift remains low enough for continuous repetition rate of the specific camera system) or rely on the purely image-based method introduced in the next section. The training data for the jet test case (4.3) were created using a time-resolved STB evaluation of an image series meant for VSC, that were captured with wind-off conditions at 10 Hz repetition rate.

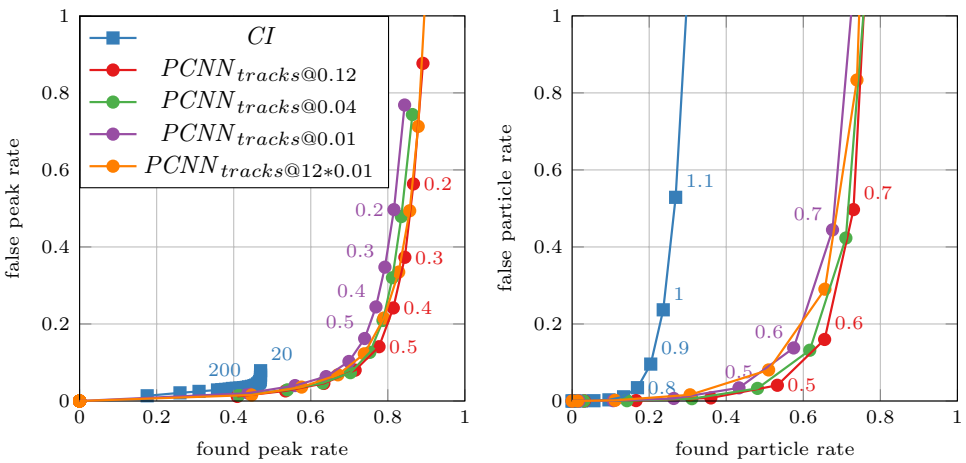


Fig. 14 Peak detection (left) and triangulation (right) performance for the 0.12 ppp RBC case of *CI* and *Peak-CNN* peak detectors. Underscore signifies that LPT results ("tracks") served as input for the *Peak-CNN* models, followed by the ppp-level of the underlying training images

5.2 Sourcing training data from low-density images

The training data generation strategies outlined so far require a pseudo ground truth in the form of tracking data. In many cases this is a feasible approach but the added complexity might be undesirable. For such situations, we introduce an additional training data generation technique which does not rely on the availability of LPT data at all. Only images at a reduced seeding density—either already available VSC data or from a short dedicated series—are required. Their seeding density should be selected such that a conventional peak detector is still able to yield reliable results (low number of overlapping particle images).

This data gained from peak detection on single images could directly be used to train a *Peak-CNN* detector. However, it cannot be expected that such a model outperforms the results that were used for its training. Following the reasoning from the last section, we therefore create a training dataset with higher perceived particle image density by stacking both the low-density images, as well as the peak detection results, until a desired density is reached. To test this approach, we take arbitrary images at 0.01 ppp from the RBC test case, apply *CI* peak detection and always stack 12 of these images to reach an effective density of 0.12 ppp. The *CI* detector would not be able to handle the final stacked image on its own well but since it is applied to the individual low seeded images the final result contains labels even for closely overlapping particle images. 30 of these stacked images together with the stacked *CI* results are used to train a $Peak-CNN_{peaks@12*0.01}$ model. For reference, we also train a $Peak-CNN_{peaks@0.01}$ model using the pure peak detection results (without stacking).

Figure 15 shows the results by comparing the performance of the two gained models on 0.12 ppp images (orange and purple curves) to *CI* (blue), as well as the *Peak-CNN* models trained with LPT results at 0.12 ppp (red) and 0.01 ppp (green). It can be seen that, as expected, training *Peak-CNN* directly with results from *CI* does not result in a model that exceeds the training data. The ground truth data does not supply correctly labeled strongly overlapping particles so the model is unable to learn how to resolve them. The stacking approach on the other hand significantly enhances the performance, even though the models trained by tracking data still hold an advantage—even the one using low-density 0.01 ppp images. This can also be seen in Fig. 16 which shows detection results of the two models on the 0.12 ppp images.

The great advantage of the stacking approach is the low effort needed to obtain the training data. As long as the required low-density images are available, no complicated processing is needed and the user interaction is limited to setting a detection threshold for *CI*. We believe this to be the most attractive pathway to bring the proposed peak detection method into practical application due to its "single click" character, performing data generation and training of the model with minimal user intervention.

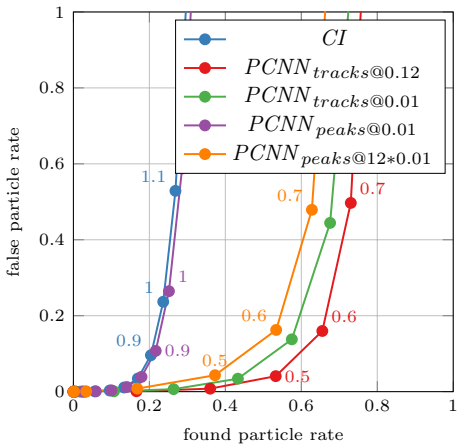


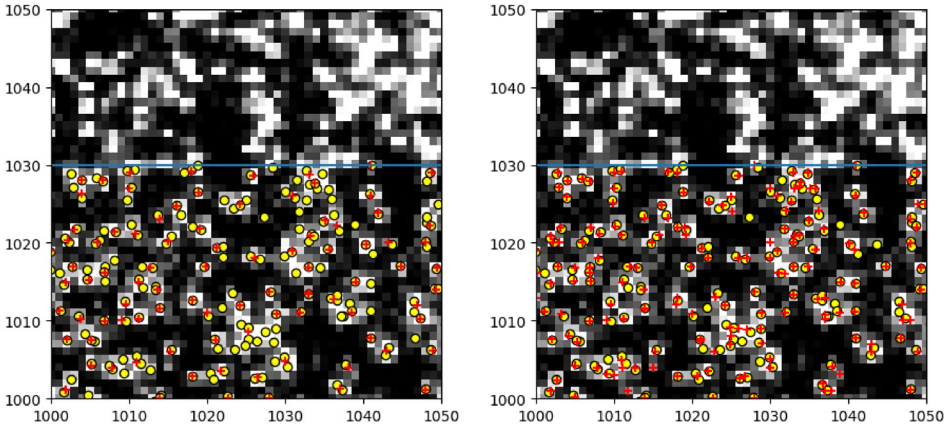
Fig. 15 Triangulation performance of *CI* and *Peak-CNN* peak detectors. Underscore signifies if LPT results ("tracks") or peak detection with *CI* ("peaks") served as training input for the *Peak-CNN* models, followed by the ppp-level of the training images

The higher quality results provided by the LPT-based labeling remain attractive for ambitious experimental setups and boundary pushing seeding densities. Often preliminary tracking results are created during setup of an experiment, which can be directly repurposed for the model training.

6 Effect on IPR performance

We have shown that the proposed *Peak-CNN* peak detector yields significantly superior results in peak detection, which directly translates into triangulation performance. These findings demonstrate the usefulness of the approach for 2D particle tracking and classical 3D-PTV algorithms that work directly with triangulation results. For these methods, the *Peak-CNN* peak detector will be a drop-in replacement, instantly yielding an increased tracking performance.

Here we go one step further and demonstrate the reconstruction performance of IPR when using *Peak-CNN* instead of *CI* for peak detection. IPR uses an iterative approach, which is tuned to attenuate the negative effects of weak peak detection performance on densely populated images. Especially with synthetic data this approach is surprisingly efficient in successively ablating the residual image, and to reconstruct more and more true particles. The first publication on IPR showed a nearly perfect reconstruction of images up to 0.05 ppp (Wieneke 2012). A number of enhancements increased the performance to 0.14 ppp (Jahn et al. 2021). To highlight the effect of *Peak-CNN* within a full IPR evaluation, we again use test cases from Jahn et al. (2021) (1300 by



(a) $Peak\text{-}CNN_{peaks@0.01}$ (b) $Peak\text{-}CNN_{peaks@12*0.01}$

Fig. 16 Models trained on 0.01 ppp data using *CI* derived labels applied to images at 0.12 ppp. Ground truth as yellow circles, detections as red crosses

1300 pixel cameras, Case I: clean images; Case II: images with moderate noise and particle intensity variations).

We embed our trained peak detector into a full IPR processing and modify the evaluation parameters of the IPR algorithm, as the parameters selected in Jahn et al. (2021) are not optimal, given the enhanced peak position accuracy offered by *Peak-CNN*. Most prominently, the allowed triangulation errors within IPR can be set lower, as shown in the previous sections. Jahn et al. (2021) applied triangulations within all iterations of IPR, where it was allowed that one of the cameras does not exhibit a found peak (only 3 cameras are required to have a peak in order to register the particle). Since the new detector finds peaks in greater number and at higher accuracy, we use this approach only in later iterations, when the number of detected peaks has decreased already. Otherwise the number of ghost particles would be unnecessarily increased. Instead, for the first five iterations all cameras are required to register a peak, while ramping the triangulation radius from 0.35px to 0.65px. This is followed by 15 iterations requiring only 3 cameras, again ramping from 0.35px to 0.6px radius. The intent is to modify some commonly adjusted IPR parameters in order to better exploit the advantages conferred by the better peak detection without departing too far from the processing used by Jahn et al. (2021). An individually optimized set of IPR parameters (Godbersen and Schröder 2023) would likely yield even better results.

The detector was trained using data at 0.05 ppp in order to simulate data generation at lower seeding densities. The training dataset is augmented with artificial residual images. Within the IPR already reconstructed particles from prior iterations are back-projected and their OTF-weighted image is subtracted from the original image resulting in the residual. This subtraction process is never fully accurate, leaving leftover artifacts like low-intensity rings (if the OTF is not wide enough or the particle images are not fully Gaussian), single pixels in the center (if the OTF is not wide enough) or crescent-shaped regions (if the particle position is still off). If the model does not encounter such marred images during training, the performance is reduced. For each image in the training dataset, we generate three artificial residual images with 50%, 10% and 1% of peaks remaining representing images at progressing IPR iterations. The other peaks are simply subtracted from the image with some noise added to their position and intensity in order to produce the aforementioned image artifacts. Since this represents a more challenging detection problem, we have increased the number of snapshots used for training to 100 instead of the 30 used before.

Figure 17 shows the results of IPR using the novel *Peak-CNN* detector for the clean Case I in terms of correctly found particles and ghosts for various ppp values. All variants use the same IPR parameters and were run for the full 20 iterations mimicking the presentation by Jahn et al. (2021). We have included their highest ppp result for comparison. Due to the strongly increased number of correctly identified particle peaks and the lower allowed triangulation error, the new evaluations are able to identify a majority of the true particles in the first iteration for all ppp variants, while the ghost fraction remains below 0.4 even for the 0.2 ppp case. The previous evaluation, using *CI*, was only able to find 19 % of the true particles, with a much higher ghost fraction of almost 0.6. The new evaluation using *Peak-CNN* at this ppp value already finds 76% of the true particles with only 17% ghosts.

The new evaluations quickly converge from the starting point, reaching > 99% of found particles after few iterations, while at the same time eliminating all ghost particles. On the other hand, the old evaluation for 0.16 ppp shows a slower convergence of true particles with the ghost particle ratio increasing over the first five time steps to nearly 1.0 and only slowly diminishing in following

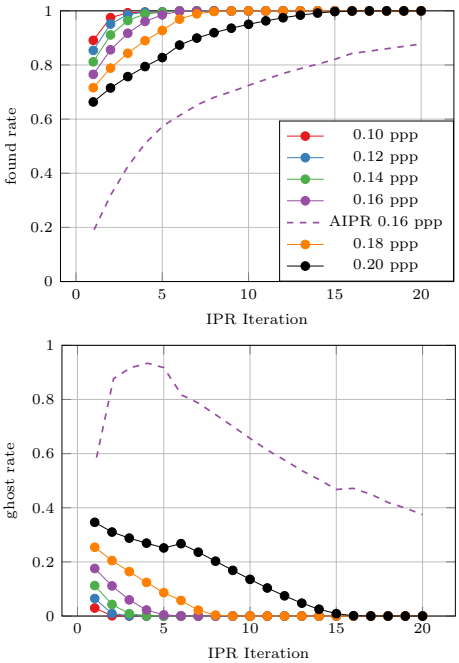


Fig. 17 IPR performance with our *Peak-CNN* detector on the clean synthetic case for various ppp values. The same model trained using only 0.05 ppp data was used for all. The dashed line represents the 0.16 ppp results from Jahn et al. (2021) using the conventional *CI* peak detector. *Peak-CNN* evaluation uses adapted IPR parameters

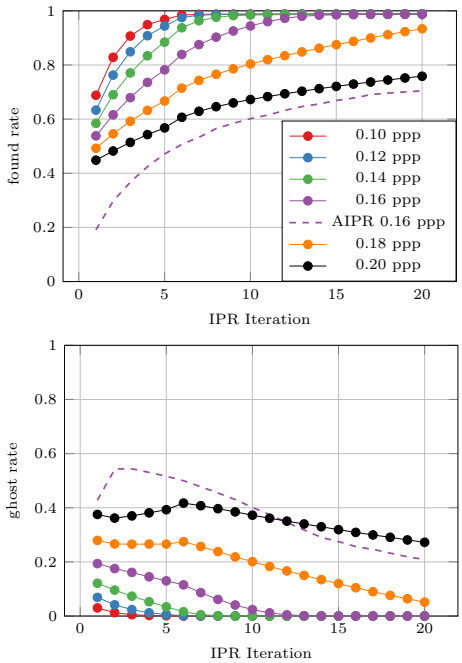


Fig. 18 IPR performance with our *Peak-CNN* detector on the noisy synthetic case for various ppp values. The same model trained using only 0.05 ppp data was used for all. The dashed line represents the 0.16 ppp results from Jahn et al. (2021) using the conventional *CI* peak detector. *Peak-CNN* evaluation uses adapted IPR parameters

iterations. No convergence to a full solution is reached within 20 iterations. In contrast our new approach is able to fully solve a 0.2 ppp case in 15 iterations. This shows that the updated peak detector is able to significantly increase the already strong performance of IPR.

Since the clean Case I represents an idealized measurement problem, we additionally conduct the same study on the more realistic Case II as shown in Fig. 18. As expected the performance decreases due to the more challenging imaging conditions. More iterations are needed for all ppp variants and less particles are found within the first iteration. Note that even for the lower seeding densities considered slightly less than 100% of all particles are recovered. This is presumably due to some particles of low brightness disappearing in the image noise and can also be observed in the results of Jahn et al. (2021). We still consider such cases fully solved.

For this more challenging problem, we are still able to fully solve the 0.16 ppp case in well under 20 iterations. The 0.18 ppp case is almost solved within these, and the 0.2 ppp case reaches 75% of particles found with an ghost fraction under 0.3. Again a strong improvement is obtained over the prior results using the *CI* detector where for 0.16 ppp only 70% of particles were obtained with 20% ghosts still remaining.

7 Conclusion and outlook

We have introduced a single-stage CNN-based approach for peak detection and compared its performance to several standard approaches. The method shows a significantly improved performance on test cases ranging from simple synthetic images to more complex real-world datasets. Its ability to better resolve overlapping particle images leads to a distinct increase of correctly identified peaks as well as improved position accuracy, compared to standard approaches. 2D particle tracking methods should directly benefit from a drop-in replacement of the currently used peak detector. Looking at triangulation performance—using a multi-camera system—the benefits are even more pronounced as the successful triangulation of a 3D particle requires a correctly identified peak on all cameras. This property will give 3D-PTV approaches, relying solely on triangulation, access to higher seeding densities than currently possible. Methods like IPR are designed to circumvent the deficits of the currently used peak detectors to a certain degree by iteratively ablating the residual image. Therefore, the gains of the CNN peak detector are not quite as pronounced when using such techniques; however, IPR still clearly benefits, allowing for higher seeding densities, fewer ghost particles and an accelerated run-time required for convergence. This will be of particular interest for

multi- or two-pulse STB applications which rely heavily on IPR performance. The exemplary dataset of a subsonic jet uses a polarization based pulse separation strategy and is thus more focused on challenging image quality. On the other hand, multi-exposure based strategies for capturing four or more pulses on double-frame cameras are more commonly used now. These methods lead to increased effective particle image densities and potential overlap of the two illuminations of slower particles, playing even more into the strengths of the new approach. At the current state, the model should be trained on a per-experiment basis. We have introduced several methods for generating training data, using either tracking results or standard peak detection on low-density images. Particularly the latter approach offers a training method with minimal additional effort, apart the recording of low-density images.

It would be highly desirable to generate a general purpose model that can be applied to new experiments with no additional training. Building such a generalized model is the focus of future work and will need adjustments to the model architecture in terms of normalization and weighting strategies as well as increased metadata input in order to cover a wide variety of experiments with diverse imaging conditions with equal performance. Part of the current model architecture is already set up with this goal in mind—therefore, generalization will be possible to a certain extent, especially for experiments taken under fairly similar conditions. Initial testing using synthetic data generated for a variety of conditions has been promising, but further study and modifications are needed. We envision an openly available model, which was trained on a number of available measurements and is continuously updated as new experiments are conducted, possibly with the aid of the community using it. However, the current state with individually trained models already provides a viable approach which ultimately will remain competitive due to the specialization on the individual experiment it is applied to.

We provide an open-source implementation of *Peak-CNN* at https://github.com/pgodb/Peak-CNN

Acknowledgements The project leading to this journal article has received funding in the frame of the project HOMER from the European Union's Horizon 2020 research and innovation program under grant agreement No. 769237. This work was partly supported by the DFG through Grant No. SCHR 1165/5-2 in the Priority Program on Turbulent Superstructures (No. SPP 1881).

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Bosbach J, Schanz D, Godbersen P, Schröder A (2021) Spatially and temporally resolved measurements of turbulent Rayleigh-Bénard convection by Lagrangian particle tracking of long-lived helium-filled soap bubbles. In: Proceedings of 14th international symposium on particle image velocimetry, vol. 1, pp. 1–12

Boomsma A, Troolin D (2018) Time-resolved particle image identification and reconstruction for volumetric 4d-ptv. In: 19th international symposium on the application of laser and imaging techniques to fluid mechanics, pp. 16–19

Carosone F, Cenedese A, Querzoli G (1995) Recognition of partially overlapped particle images using the Kohonen neural network. Exp Fluid 19(4):225–232

Cheminet A, Krawczynski J-F, Druault P (2018) Particle image reconstruction for particle detection in particle tracking velocimetry. Measure Sci Technol 29(12):125202. https://doi.org/10.1088/1361-6501/aae9c2

Cardwell ND, Vlachos PP, Thole KA (2011) A multi-parametric particle-pairing algorithm for particle tracking in single and multiphase flows. Measure Sci Technol 22(10):105406. https://doi.org/10.1088/0957-0233/22/10/105406

Dabiri D, Pecora C (2019) Particle tracking velocimetry. IOP Publish. https://doi.org/10.1088/978-0-7503-2203-4

Elsinga G, Westerweel J, Scarano F, Novara M (2011) On the velocity of ghost particles and the bias errors in tomographic-piv. Exp Fluid 50(4):825–838

Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Gordon, G., Dunson, D., Dudík, M. (eds.) Proceedings of the fourteenth international conference on artificial intelligence and statistics proceedings of machine learning research, PMLR, Fort Lauderdale, FL pp. 315–323

Godbersen P, Schröder A (2023) Automated parameter optimization for LPT measurements using multi-objective evolutionary algorithms. In: 15th International Symposium on Particle Image Velocimetry 2023, vol. 256, pp. 1–9

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML'15, pp. 448–456.

Jahn T, Schanz D, Schröder A (2021) Advanced iterative particle reconstruction for Lagrangian particle tracking. Exp. Fluid 62(8):1–24

Kingma DP, Ba J (2017) Adam: A method for stochastic optimization.

König J, Chen M, Rösing W, Boho D, Mäder P, Cierpka C (2020) On the use of a cascaded convolutional neural network for three-dimensional flow measurements using astigmatic ptv. Measure Sci Technol 31(7):074015

Liang J, Liu X, Chen T, Pan C, Xu C (2022) Particle identification in particle tracking velocimetry using two-stage neural networks. J Indus Manage Optim 19:5331–5352

Maas HG, Papantoniou D, Gruen A (1993) Particle tracking velocimetry in three-dimensional flows: Part 1. Photogrammetric determination of particle coordinates. Exp Fluid 15(2):133–146

Meller Y, Liberzon A, Segal M, Zandbank I, daniellaillouz devowit Corbetta A, Lüthi B (2019) Openptv/openptv: Updated

the version number 0.2.6 in the setup.py https://doi.org/10.5281/zenodo.3229411

Manovski P, Novara M, Mohan NKD, Geisler R, Schanz D, Agocs J, Godbersen P, Schröder A (2021) 3d Lagrangian particle tracking of a subsonic jet using multi-pulse shake-the-box. Exp Therm Fluid Sci 123:110346

Mann J, Ott S, Andersen JS (1999) Experimental study of relative, turbulent diffusion. Risø National Laboratory.

Novara M, Schanz D, Geisler R, Gesemann S, Voss C, Schröder A (2019) Multi-exposed recordings for 3D Lagrangian particle tracking with multi-pulse shake-the-box. Exp Fluid 60(3):44. https://doi.org/10.1007/s00348-019-2692-7

Novara M, Schanz D, Schröder A (2023) Two-pulse 3D particle tracking with shake-the-box. Exp Fluid 64(5):93. https://doi.org/10.1007/s00348-023-03634-7

Ouellette NT, Xu H, Bodenschatz E (2006) A quantitative study of three-dimensional Lagrangian particle tracking algorithms. Exp Fluid 40(2):301–313

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: international conference on medical image computing and computer-assisted intervention, pp. 234–241 . Springer

Sax C, Dreisbach M, Leister R, Kriegseis J (2023) Deep learning and hybrid approach for particle detection in defocusing particle tracking velocimetry. Measure Sci Technol 34(9):095909

Schanz D, Gesemann S, Schröder A, Wieneke B, Novara M (2012) Non-uniform optical transfer functions in particle imaging: calibration and application to tomographic reconstruction. Measure Sci Technol 24(2):024009. https://doi.org/10.1088/0957-0233/24/2/024009

Schanz D, Gesemann S, Schröder A (2016) Shake-the-box: Lagrangian particle tracking at high particle image densities. Exp Fluid 57(5):70. https://doi.org/10.1007/s00348-016-2157-1

Sciacchitano A, Leclaire B, Schröder A (2021) Main results of the first Lagrangian particle tracking challenge. In: 14th international symposium on particle image velocimetry.

Schröder A, Schanz D (2023) 3D Lagrangian particle tracking in fluid mechanics. Ann Rev Fluid Mech 55(1):511–540. https://doi.org/10.1146/annurev-fluid-031822-041721

Wieneke B (2012) Iterative reconstruction of volumetric particle distribution. Measure Sci Technol 24(2):024008. https://doi.org/10.1088/0957-0233/24/2/024008