Utilising Convolutional Neural Network Algorithms For X-Ray Image Analysis Of A Fluidised Bed

Peter Laszcz

August 2024

1 Introduction

As part of a MEng project in 2022, a series of x-ray photos were taken of a functioning fluidised bed of sand injected with varying loads of silicon oil. The aim was to apply image analysis in order to validate both PEPT results and other findings (such as phase behaviour). However, due to time constraints, all x-ray analysis was carried out qualitatively by eye, and therefore no method was developed to carry out quantitative analysis of bubbles and jets in the fluidised bed. This project aims to develop such a method in order to realise the potential for quantitative analysis of fluidised beds through radiography.

2 Developing a Method

A basic approach to carrying out quantitative analysis would be to process the images using techniques such as thresholding (binary image filtering through raw pixel values), then contouring the resulting binary image using a tool such as OpenCV. However, this has some major drawbacks; due to the low resolution of the cropped images (400x400), thresholding is highly destructive, especially to objects which often appear very dim on the image (such as jets)- this not only reduces the number of data points but also undermines validity on gathered data such as jet length and bubble area. Another issue with using OpenCV is that, as blurry images ted to "fuse" clusters of bubbles and jets together, it is impossible to segment these clusters apart; the image processing techniques of opening and closing prove minimally effective. It is also difficult to classify the binary images into bubbles and jets, meaning that jet data could be introduced into bubble data and vice versa. Utilising a Convolutional Neural Network (or CNN) for the computer vision task eliminates these issues; a trained model is able to both segment and classify bubbles and jets with satisfactory speed and accuracy, without destructive image processing. Furthermore, the segmentation masks are easily exportable to OpenCV, which allows us to combine the segmentation and classification accuracy of a CNN with the analysis tools of OpenCV to produce a full and accurate dataset.

3 Method

The CNN algorithm chosen for the job was the YOLO (You Only Look Once) algorithm, due to its respectable accuracy and high speed, which was important due to time constraints. It also has a large user base, meaning that modifying or troubleshooting the system will be easier given the vast quantity of available documentations. YOLO v8 was picked because, at the time of writing, it is the most accurate version to support image segmentation. The training dataset was formed of 500 images, with about 39 random images per run. To provide annotations that were as accurate as possible, the x-rays were loaded into OpenCV and had their backgrounds "subtracted" using a photo of the empty bed. This allowed the bubbles and jets to be isolated from the surroundings. The annotation was carried out on Roboflow Annotate; whilst other open source options with less restrictive licenses are available, such as CVAT, the annotation tool was picked for its annotation using Meta's Segment Anything Model (or SAM). The application of SAM towards training data provided the speed needed for one person to comfortably carry out the annotation and remainder of the method in the set timeframe (7 days), as it allowed most item boundaries to be drawn automatically. 1981 jets and 1270 bubbles were annotated, with a split of 70% training, 20% validation, and 10% testing. The dataset was then downloaded and used to refine the YOLO v8 segmentation model. The largest variant, YOLO v8x-seg, was used as its increased accuracy outweighed the increase in time for inference:

Model	Size (pixels)	mAP^{box}	mAP^{mask}	Speed w/A100 GPU (ms)
YOLOv8n-seg	640	36.7	30.5	1.21
YOLOv8s-seg	640	44.6	36.8	1.47
YOLOv8m-seg	640	49.9	40.8	2.18
YOLOv8l-seg	640	52.3	42.6	2.79
YOLOv8x-seg	640	53.4	43.4	4.02

Table 1: Table from Ultralytics, with sample metrics for each variant (where mAP is the Mean Average Precision of inference). The local performance comparison of different models was predictable from this as our inference runs on an A100 GPU.

Training and inference took place utilising an A100 node on BlueBEAR, as running the system on CPU was too slow for the set time constraints. With the GPU involved, inference runs at 12.8ms per image.

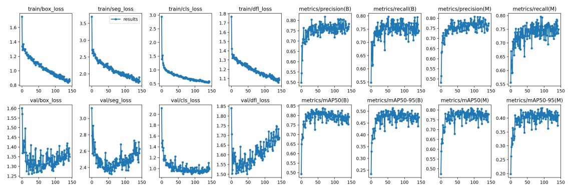

Figure 1: Performance data from training.

Running prediction on an image using the refined model produced satisfactory results:

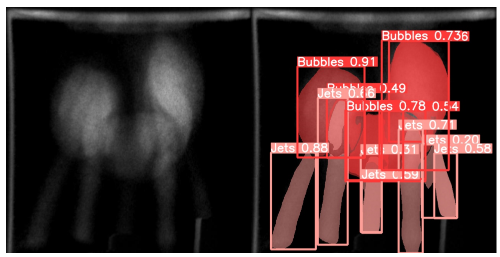

Figure 2: Image before and after inference.

As shown in Figure 2, the algorithm will pick apart clusters of bubbles and tends not to confuse the 2 item classes (bubbles and jets). Unfortunately, one factor that the algorithm will not account for is the presence of black squares in the image as unit markers:

Figure 3: Empty bed.

This may reduce the jet length validity of this particular run by obscuring the bottom section of some jets. The segmentation masks were extractable into a NumPy array, which OpenCV recognises. Filtering the extraction by mask class allows one to run operations on exclusively bubbles or exclusively jets.

Another data point of interest was the velocity of the bubbles, as this would allow one to examine the relationship between PEPT jump times and actual bubble velocity. In an attempt to achieve this, the tracking library of YOLO v8 was used, by loading each run into a video file and then feeding the video into the trained model. A frame-by-frame approach produced more successful results than whole-video processing, but said results were still unusable, being unable to track bubbles over multiple frames. One reason for this is, whilst the x-ray camera at University College London is capable of 72 frames per second, the experiment only ran at 36 frames per second- doubling the frame rate would greatly enhance tracking capabilities as, at the frame rate currently used, a bubble will only remain in the image for around 2 frames. The dimensions of the bed also limit the capability of tracking; an H/D of 1 means that a bubble travels a relatively short distance before leaving the bed- this narrows the window of opportunity for tracking. Finding a better method, perhaps using the recently announced Segment Anything Model v2 (or SAM v2), should be a priority for anyone looking to study bubble velocities in greater detail using x-ray radiography.

4 Results

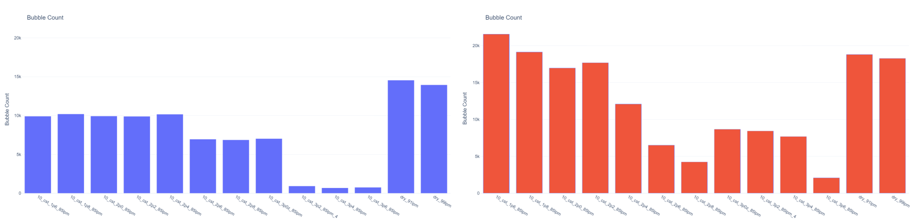

Figure 4: Detected bubbles with OpenCV (left) and YOLO v8 (right) methods.

Overall, the YOLO v8 algorithm delivers a 59.45% higher rate of success in bubble detection than its OpenCV counterpart (n.b that this is an overall percentage of total bubbles detected), resulting in fuller datasets. Furthermore, the lack of destructive method allows for much more accurate measurements to be carried out on both bubbles and jets.

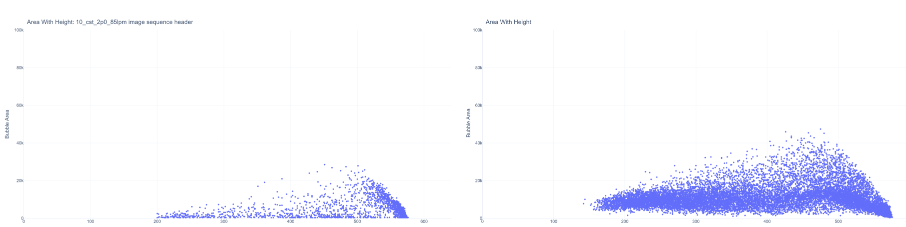

Figure 5: Bubble area/height graph using OpenCV (left) and YOLO v8 (right).

A select range of data was produced, including bubble area distribution histograms, distribution histograms of jet length, bubble count charts, and plots of area with height. The scripts of these demonstrate the cross compatibility of a segmentation CNN with measuring tools within the OpenCV toolbox, allowing pre-existing code to be easily modified to adopt said model.

5 Conclusion

The utilisation of CNNs for x-ray computer vision in fluidised beds is a promising alternative to traditional image processing as it offers both minimal data loss and a higher level of nuance in distinguishing and segmenting bubbles and jets. A Mean Average Precision of approx. 80% is a good start, especially as most inaccuracies come from item loss and not class confusion, but an improvement to 90% or higher would be both achievable and desirable for future deployment.

6 Future Work

For future improvement, a user with less time constraints should consider SAM v2, or even the older Mask-R CNN, to rewrite the base image segmentation side. Either of these other models would offer a significant accuracy increase, but would take longer to develop. SAM v2 would also yield a great improvement in the tracking of bubbles owing to its specialisation in segmentation tracking. A larger team should consider collaboratively annotating a larger training dataset, as the size of the current dataset may hinder further improvement to Mean Average Precision.

Development of good tracking would also ease the validation of Convolutional Neural Networks as a method, because bubble diameter could be curve fit with velocity using the Darton bubble velocity equation.

7 Acknowledgements

- The BlueBEAR HPC facility.
- The original MEng project, "Development of Fluidized Bed Systems for Innovative Recycling Methodologies" by Franklin Elsim Putra and Nikita Simarmata.
- University College London for providing X-ray radiography images.
- Kit Windows-Yule for supervising the project.
- Dominik Werner and Niklas Adio for providing help throughout the project.
- Ander Hatton for proofreading this writeup.