materials

MDPI

Article

Deep-Learning-Based Segmentation of Keyhole in In-Situ X-ray Imaging of Laser Powder Bed Fusion

William Dong [1], Jason Lian [2], Chengpo Yan [2], Yiran Zhong [2], Sumanth Karnati [2], Qilin Guo [1], Lianyi Chen [1,3,*] and Dane Morgan [3,*]

[1] Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA; williamdong@berkeley.edu (W.D.); qilin.guo@unl.edu (Q.G.)
[2] Department of Computer Science, University of Wisconsin-Madison, Madison, WI 53706, USA; jlian7@wisc.edu (J.L.); cyan46@wisc.edu (C.Y.); yzhong68@wisc.edu (Y.Z.); karnati2@wisc.edu (S.K.)
[3] Department of Material Science & Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA
* Correspondence: lianyi.chen@wisc.edu (L.C.); ddmorgan@wisc.edu (D.M.)

Abstract: In laser powder bed fusion processes, keyholes are the gaseous cavities formed where laser interacts with metal, and their morphologies play an important role in defect formation and the final product quality. The in-situ X-ray imaging technique can monitor the keyhole dynamics from the side and capture keyhole shapes in the X-ray image stream. Keyhole shapes in X-ray images are then often labeled by humans for analysis, which increasingly involves attempting to correlate keyhole shapes with defects using machine learning. However, such labeling is tedious, time-consuming, error-prone, and cannot be scaled to large data sets. To use keyhole shapes more readily as the input to machine learning methods, an automatic tool to identify keyhole regions is desirable. In this paper, a deep-learning-based computer vision tool that can automatically segment keyhole shapes out of X-ray images is presented. The pipeline contains a filtering method and an implementation of the BASNet deep learning model to semantically segment the keyhole morphologies out of X-ray images. The presented tool shows promising average accuracy of 91.24% for keyhole area, and 92.81% for boundary shape, for a range of test dataset conditions in Al6061 (and one AliSi10Mg) alloys, with 300 training images/labels and 100 testing images for each trial. Prospective users may apply the presently trained tool or a retrained version following the approach used here to automatically label keyhole shapes in large image sets.

Keywords: keyhole; laser powder bed fusion; deep learning; image segmentation

check for updates

Citation: Dong, W.; Lian, J.; Yan, C.; Zhong, Y.; Karnati, S.; Guo, Q.; Chen, L.; Morgan, D. Deep-Learning-Based Segmentation of Keyhole in In-Situ X-ray Imaging of Laser Powder Bed Fusion. *Materials* **2024**, *17*, 510. https://doi.org/10.3390/ma17020510

Academic Editors: Tuhin Mukherjee and Qianru Wu

Received: 16 October 2023
Revised: 10 January 2024
Accepted: 16 January 2024
Published: 21 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Laser powder bed fusion (LPBF), also known as selective laser melting (SLM), is currently one of the most common metal additive manufacturing (AM) techniques [1,2]. During the LPBF process, a focused laser will shoot onto the powder bed selectively and make powders melt, merge, and solidify to build up a part based on the CAD (computer-aided design) model [3]. Using in-situ X-ray imaging to monitor the process, previous studies have found that high intensity of the laser will result in vaporization of the material, which will lead to recoil pressure that pushes the molten metal in the melt pool to create a vapor cavity, named a 'keyhole' [4,5]. In the process and along the printing path, the keyhole experiences inconsistent recoil pressure, surface tension, and Marangoni force, leading to severe and random fluctuations [6,7]. Sometimes, fluctuated keyholes will collapse, and the vapor will be partially trapped inside the melt pool, resulting in undesirable porosity in the final product [4,8].

Many studies have been made to find correlations between the keyhole morphology revealed in the X-ray imaging and the generation of keyhole-induced pores [9,10], with an eventual goal of reducing defects. Due to the complex dynamics of the LPBF process, there is great interest in in-situ data-driven methods to study the defect formation

mechanisms [11,12]. As a popular approach to observe the dynamics in the LPBF process, in-situ X-ray imaging can be captured at a frame rate of over one million frames per second, generating enormous keyhole morphology images, and creating a huge potential for data-driven studies with keyholes. However, these studies are greatly constrained by a limited quantity of well-characterized keyhole morphology data, which typically needs to be labeled by humans. More specifically, sufficient data for machine learning-based pore prediction for many types of LPBF systems, conditions, and alloys, will require very large efforts to label keyhole morphologies unless accessible automatic tools that can segment the keyholes are developed. Several automatic tools have recently been explored in previous works: Pyeon et al. developed a non-machine learning-based semi-automatic keyhole region extraction tool [13], and Zhang et al. tested several semantic segmentation and object detection models and compared the performances of extracting keyholes and pores at the same time [14]. However, the filter developed by Pyeon et al. was only tested with clean images without metal powder, and models tested by Zhang et al. segment both keyholes and pores in the same classification. In addition, while the boundary is the most important feature in the keyhole morphology, previously proposed methods are not designed to have high segmentation boundary accuracy and be validated against datasets from different experiments. Therefore, considering the increasing need for keyhole segmentation for large X-ray imaging databases, we here develop an automatic keyhole segmentation tool with high accuracy for both area and boundary and test against datasets with powders from different experiments.

In this paper, a deep-learning-based semantic segmentation tool that is capable of automatic segmentation of keyholes in X-ray images with accurate boundaries was developed. This tool is composed of a filter that standardizes, normalizes, and cleans the X-ray images, and an implementation of BASNet, a Boundary-Aware Segmentation network, that predicts semantic labels [15]. Without any human inputs, this tool only requires users to run algorithms with their X-ray images, which significantly accelerates the keyhole morphology labeling process and enables the possibility of data-driven analysis with a large quantity of morphology data. In the following, the implementation of the method will be illustrated, the performance of the tool will be quantified, and the mechanism and limitation of the method will be discussed at the end. This work was conducted with data derived from different experiments at multiple different times, but the tool provided in this study will likely need to be refit for systems with significant differences from those studied here. However, such refitting can likely be performed quickly through transfer learning, i.e., by starting from the weights in this paper. Development of similar segmentation tasks for X-ray imaging, like segmenting melt pools or spatters in the LPBF process, could also be accelerated by transfer learning from the present model.

2. Methods

The workflow of the whole segmentation process is shown in Figure 1: Raw X-ray images are processed with a designed filter and then fed into the segmentation network, which outputs the semantic labels. The segmentation network needs to be previously trained by filtered images and ground truth labels in the training set.

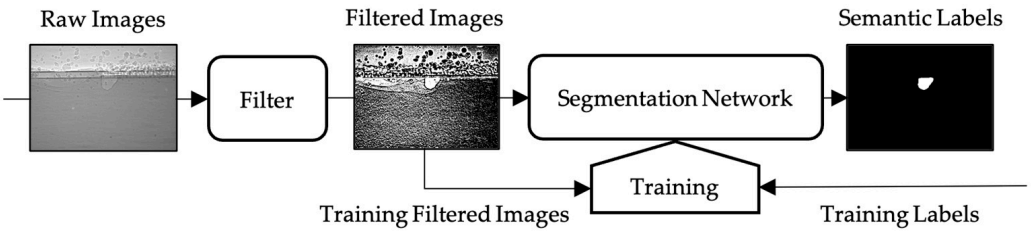


Figure 1. A flowchart illustrating the keyhole segmentation process pipeline.

2.1. Raw Images

Data used in this study were acquired by using in situ X-ray imaging on LPBF processes at the beamline 32-ID-B of the Advanced Photon Source at Argonne National Laboratory. During experiments, the ytterbium fiber laser generated a laser beam and was directed by a galvo scan head toward the metal powder and substrate. While the laser was melting the material, the scanning area was penetrated by X-ray simultaneously, and the shapes of keyholes and pores were projected and converted to visible light by a scintillator, which was recorded by a high-speed camera with a frame rate of 50 kHz [4]. There are 8 X-ray imaging data sets in this work and they were acquired by 8 separate experiments, each with different processing parameters, as shown in Table 1. The experiments were performed with Al6061 in 7 cases and AliSi10Mg in one case. Each data set has 400–500 frames, and 50 of those were labeled in each data set and used in this study. In total, there are 3416 images in the 8 data sets from which 1441 images have a visible keyhole, and 400 images with visible keyholes were labeled for training and testing. The remaining 1975 images show no keyholes since the laser in the LPBF process was not on or in the field of view at the time of imaging, and are not used in the testing and training of this paper.

Table 1. Samples and processing parameters of in situ X-ray imaging experiments.

Experiment	Material	Nanoparticle	Substrate	Power (W)	Scan Speed (m/s)
1	Al6061	10%vol TiC	Printed	385	0.3
2	Al6061	10%vol TiC	Printed	385	0.4
3	Al6061	/	Printed	443	0.4
4	Al6061	/	As-cast	500	0.2
5	Al6061	/	Printed	500	0.4
6	Al6061	/	As-cast	500	0.4
7	Al6061	/	As-cast	500	0.4
8	AlSi10Mg	/	As-cast	500	0.5

2.2. Filtering

There are two major goals when filtering the raw images: (1) standardize images to be acceptable for the segmentation network as inputs, and (2) normalize and clean the image by reducing the differences between datasets and removing stationary obstructions. The filter is built using MATLAB R2023b based on the schematics shown below in Figure 2 and described below.

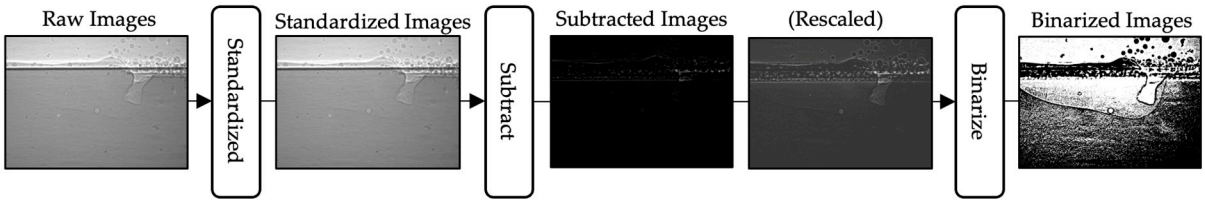


Figure 2. Flowchart illustrating filtering procedures. The subtracted image is rescaled for better representation in the following image, no re-scaling was involved in the "Subtract" and "Binarize" steps.

To meet the requirements of the network, all images need to be at the same resolution, a fixed size, and in the same format. In our case, all images are converted into uint 8, with a size of 700 × 500 pixels, and in PNG format. This step is known as "Standardize".

In different experiment setups, X-ray images will have different brightness, contrast, and sometimes stationary obstructions overlapping with the keyhole area. These factors greatly hinder the segmentation of keyholes. To alleviate these factors, a concurrent subtraction step is designed, and the detailed mechanism will be covered in the Discussion section. For each image, the sum of previous N frames of images is calculated, and the current image is subtracted by the average of previous N frames, where N is a tunable parameter for each dataset, and here 40 previous images are used. This step is termed "Subtract".

To further normalize the images, greyscale images are converted into black and white (binary) images. The images are binarized with a threshold of 0.5. The global average value of all pixels in the whole image is calculated, pixels with a value above the average are marked as 1, and pixels with a value below average are marked as 0. This step is "Binarize".

With these three steps, any raw X-ray images can be transformed into the format used for training and can be input into our training model. Please check the GitHub repository in the Supplementary Material to find a specific implementation of these steps.

2.3. Segmentation Network Training

Before the filtered images are fed into the segmentation network, the network needs to be trained by filtered training images and corresponding ground truth labeling. The ground truths are labeled manually using MATLAB R2023b Image Labeler app, with pixels assigned as 1 in the keyhole region and pixels assigned as 0 in the background, exported with the same resolution, size, and format as training images, creating a binary label for each corresponding image. In this study, experiments/datasets 3 to 8 were picked as training datasets, and their corresponding ground truth labels were picked as the training set, with 50 images and labels for each of the 6 datasets, and 300 images and labels in total.

In this study, we chose the Boundary-Aware Segmentation Network (BASNet) developed by Qin et al. as the semantic segmentation network [15]. The BASNet model was trained at a batch size equal to 1, and 70 epochs, leading to 21,000 iterations in total, and the trained model can be found in the GitHub repository in the Supplementary Material. The model appeared to be well-converged by this number of epochs and the loss curve can be found in the Results and Discussion section. The source code for training was modified to add a testing step with datasets 1 and 2 after each epoch, where the model was tested to generate the labels for images in datasets 1 and 2. The labels were compared with the ground truths and the testing losses for these two datasets were also calculated after each epoch of training. Convergence on testing loss was also observed, with more details in the Results and Discussion section.

2.4. Deep Learning Segmentation on Test Data

Experiments/datasets 1 and 2 were picked as the testing datasets for the segmentation model, and the ground truth was labeled in advance as a comparison to prediction. Filtered X-ray images for datasets 1 and 2 were fed into the trained BASNet model to generate predicted labels, which need to be binarized to convert the label from gradient to binary images. The sample X-ray image, its predicted label, and comparisons with ground truth are shown in Figure 3.

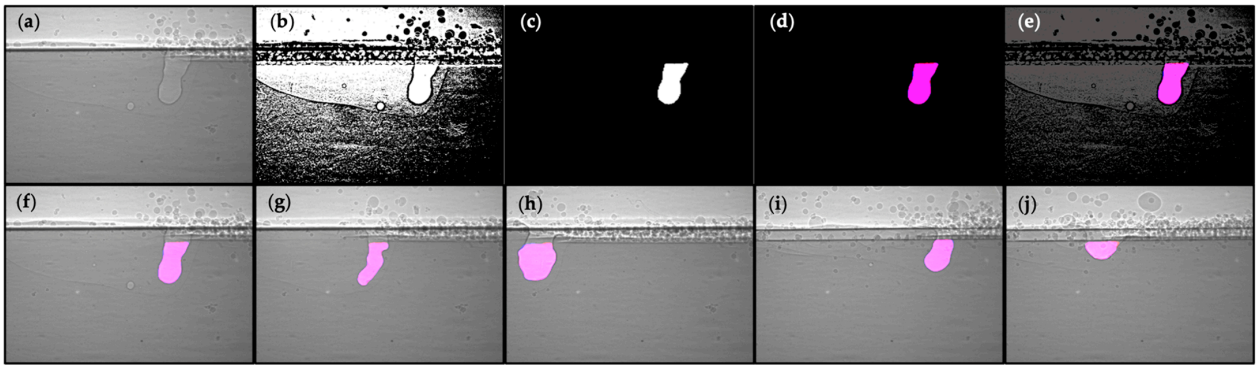


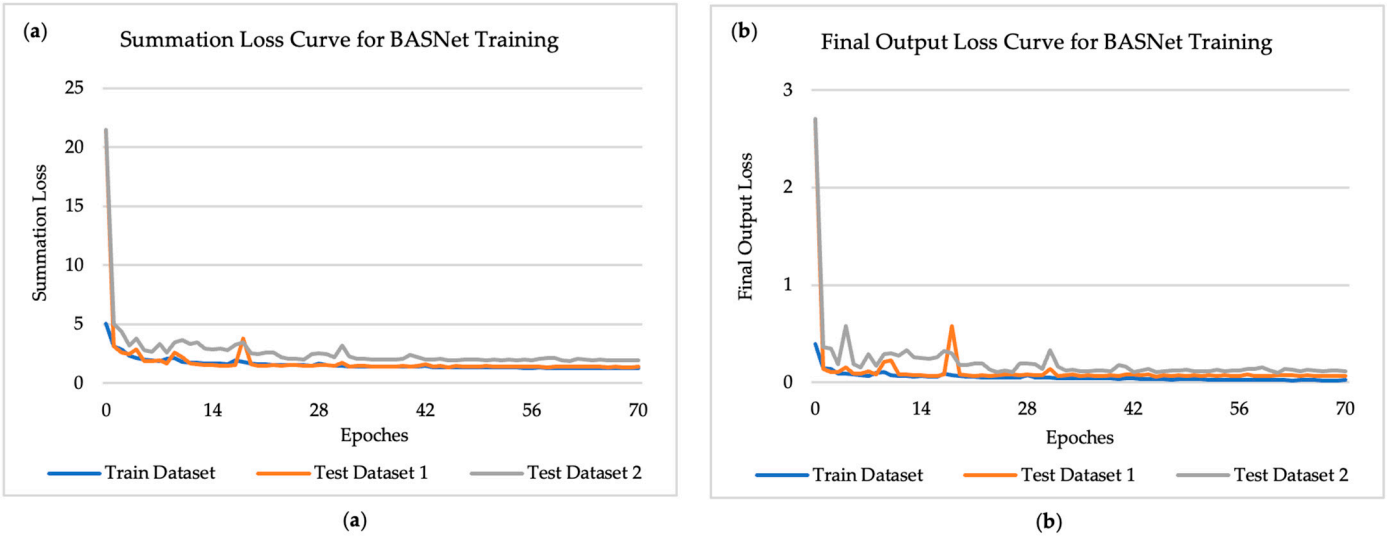
Figure 3. Sample X-ray image, its predicted label, and comparisons: (**a**) raw image; (**b**) filtered image; (**c**) predicted label; (**d**) overlay of predicted label and ground truth, where predicted label is in red, ground truth is in blue, intersection is overlapped as purple (Since there are not much unmatched predicted label and ground truth, red and blue area are barely visible for the given images); (**e**) overlay on filtered image; (**f**) overlay on raw image; (**g**,**h**) overlay on raw images for other two frames in testing dataset 1; (**i**,**j**) overlay on raw images for two frames in testing dataset 2.

Testing on data from different experiments can allow for evaluation of the actual performance of this tool when prospective users implement this tool to acquire keyhole morphology on their own X-ray images. The performance of this model on testing datasets will be quantified in the Results section.

3. Results

3.1. Training and Loss Function

The BASNet model adopts a predict-refinement structure, where the input image first passes through a predict module (encoder–decoder) and then a residual refinement module to finally generate the segmentation. The segmentation generated by the refinement module is the final output of the model, and there are 7 "side outputs", or intermediate outputs, which are the outputs of every stage of the decoder in the predict module, and are also the inputs of their upcoming stage. The loss function for optimization of BASNet takes the summation of loss values of all final outputs and 7 side outputs generated along the network, which is named "summation loss". While this is useful for training, the performance a user cares about is the loss from the final output, which is named "final output loss", as that is what will be used in applications. Both training summation loss and the final output loss for the training and test data were recorded at the end of each epoch, shown in Figure 4 below. As the training loss curve shows in Figure 4a, the model gradually converged to a low and consistent summation loss value as training proceeded. The model also performed nearly as well on test datasets, which also gradually converged to a low loss value along with the training loss. During training, the final output loss for the training datasets and test datasets was also calculated and is shown in Figure 4b below. The convergence of the training and test datasets onto low loss values indicates the high accuracy of segmentation of the trained model on both training and test datasets.



(**a**) (**b**)

Figure 4. Recorded loss curve for BASNet training process for training datasets (experiment 3–8), test dataset 1 and 2, from 0 epoch to 70 epochs: (**a**) training and testing data loss curve for summation loss; (**b**) training and testing data loss curve for final output loss.

3.2. Testing and Performance Matrices

To evaluate the performance of the tool, by comparing with the predicted label and ground truth, two matrices are calculated to quantify the segmentation accuracy of the pipeline: intersection over union (IoU) and boundary F-score (BF-score) (both defined below). Both IoU and BF-score are in the range [0, 1], an IoU closer to 1 means a better match in area, and a BF-score close to 1 means a better match on the boundary [16]. These two metrics are calculated based on the predicted label and ground truth for all images in both testing datasets 1 and 2, based on Equations (1) to (4) below, where PL and PLB stand for the model-predicted label

and its boundary, and GT and GTB stand for the ground truth label and its boundary, which represents the actual keyhole region. In the IoU calculation, the intersection area and union area of the predicted label and ground truth are calculated, and a ratio of intersection over union that is close to 1 shows both successful coverage of the actual keyhole area by the model predicted labeling and little overestimation of the actual keyhole area by the predicted labeling. In the BF-score calculation, precision represents the percentage of the model-predicted boundary that matches the actual keyhole boundary, and high precision means the model predicted a more correct boundary. Recall represents the percentage of the actual keyhole boundary that is predicted correctly by the model-predicted boundary, and high recall indicates more actual keyhole boundary is successfully predicted by the model. The BF-score is calculated by the multiplication over the sum of the precision and recall times by 2, and a score close to 1 indicates both high precision and recall by the model prediction. In Equations (2) and (3), the threshold is set for the maximum distance between two boundaries is 1 pixel, meaning that any portion of the boundary that exceeds 1 pixel distance from the other will not count in the numerator. This is a demanding criterion, representing just a fraction of a percent of the dimensions of the filtered images, which in this study are 700 by 300 pixels.

$$\text{IoU} = (\text{Area Intersection of PL and GT})/(\text{Area Union of PL and GT}) \quad (1)$$

$$\text{Precision} = (\text{Portion of PLB with distance to GTB within the threshold})/(\text{Full PLB}) \quad (2)$$

$$\text{Recall} = (\text{Portion of GTB with distance to PLB within the threshold})/(\text{Full GTB}) \quad (3)$$

$$\text{BF-score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \quad (4)$$

As shown in Table 2 below, IoU and BF-score are high and close to 1 for both datasets 1 and 2 for run 1, which means that this method successfully segments out the keyhole region in test datasets. Considering IoU, this method is on the same level as other segmentation tools proposed in previous works, and a high BF-score further validates the segmentation accuracy on the boundary [14]. The same tool is later tested using cross-validation, with three more runs trained and tested as Table 3 below. The IoU and BF-score for cross-validation are also shown in Table 2, with most testing datasets showing similar values for both matrices, with an average IoU of 0.9124 and average BF-score of 0.9281, suggesting that this tool is very accurate for random 75% training/25% test splits. However, the proposed tool might also encounter segmentation errors, as testing dataset 3 for run 2 shows relatively lower IoU and BF-score values, which will be further covered in the Discussion section.

Table 2. IoU and BF-score for cross validation with 6 training sets and 2 testing sets, 4 runs in total.

Run	Testing Dataset	IoU	BF-Score
1	1	0.9381	0.9514
1	2	0.8923	0.9098
2	3	0.8333	0.8603
2	4	0.9321	0.9421
3	5	0.9195	0.9352
3	6	0.9190	0.9351
4	7	0.9130	0.9313
4	8	0.9518	0.9595
Average		**0.9124**	**0.9281**

Table 3. Cross validation training and testing datasets assignments.

Run	Dataset 1 and 2	Dataset 3 and 4	Dataset 5 and 6	Dataset 7 and 8
1	Test	Train	Train	Train
2	Train	Test	Train	Train
3	Train	Train	Test	Train
4	Train	Train	Train	Test

4. Discussion

We constructed a pipeline for keyhole region semantic segmentation in in-situ LPBF X-ray Imaging. The whole pipeline consists of two main components, a filter that standardizes, normalizes, and cleans raw datasets, and deep learning segmentation labels the keyhole region from the filtered images. In different experiment setups, X-ray images have different brightness, contrast, and sometimes stationary obstructions overlapping with the keyhole area. These factors greatly hinder the segmentation of keyholes.

Therefore, the intuition of the design of the filter part in the pipeline is to manage all training data in a consistent fashion, so as to reduce the effects of attributes of the datasets on the prediction of keyhole segmentation. From the eight raw datasets used in this experiment, images from four datasets have a dimension of 712 × 512 pixels, and images from four other datasets have a dimension of 896 × 448 pixels, all in TIF format. Theoretically, having images with different sizes should still be applicable for training, as there is a rescaling step in the training algorithm, but a consistent size and file format will make image labeling and manipulation much easier. Hence, the designed Standardize step converts all images into 700 × 500 pixels and PNG format.

As shown in Table 4, raw images from different datasets have vastly different brightness and contrast values, and the differences can be mitigated with the Subtract and the Binarize step to normalize images from all datasets. Firstly, the Subtract step greatly reduces the differences by subtracting the average image of the whole dataset, leaving only the features of each image relative to the dataset. Then, the differences are further alleviated by the Binarize step, which reduces the greyscale difference across different datasets by turning the image into black and white, which will have contrast of 1.

Table 4. Average brightness and average contrast value for raw, subtracted, and binarized images for all images in 8 datasets. For each image, brightness is calculated by the mean of all pixels' value over the white value (255 for uint 8), and contrast is calculated by the range of pixels' value (maximum − minimum) over the white value. Values from all 50 images for each dataset are averaged.

Dataset	Average Brightness			Average Contrast		
	Raw	Subtracted	Binarized	Raw	Subtracted	Binarized
1	0.5852	0.0079	0.0363	0.8034	0.4820	1.0000
2	0.5802	0.0091	0.0374	0.7496	0.4958	1.0000
3	0.0131	0.0001	0.0314	0.0278	0.0155	1.0000
4	0.0113	0.0001	0.0275	0.0255	0.0129	1.0000
5	0.0131	0.0001	0.0328	0.0288	0.0156	1.0000
6	0.5841	0.0097	0.0350	0.7833	0.5260	1.0000
7	0.0115	0.0001	0.0215	0.0264	0.0119	1.0000
8	0.5876	0.0083	0.0389	0.7503	0.4261	1.0000

Despite brightness and contrast, stationary obstruction also greatly hinders the segmentation of keyholes, especially when they are spatially overlapped with the keyhole area. These obstructions are stationary and are affiliated with a particular dataset, which yields inconsistent X-ray images across different datasets and influences the prediction of the keyhole area. As shown in Figure 5, the obstruction can also be resolved with the Subtraction step, as the subtraction value shown in Figure 5b contains the stationary patterns, and stationary obstructions can be removed for subtracted and binarized images, as shown in Figure 5c–e. Note that the deep learning model segmentations described below with just the Standardize step and just the Subtract step were also tried, but the model did not perform well and failed to identify the keyhole region in most frames.

For the deep learning segmentation, BASNet utilizes a hybrid fusion loss function to achieve training supervision of prediction on multiple levels: pixel level, patch level, and map level, instead of barely relying on IoU, which could lead to insufficient prediction of structural properties on the patch level. Along with the prediction-refinement structure, BASNet is capable of semantic segmentation for boundary-sensitive cases like the detailed morphology in keyhole segmentation. The boundary prediction performance is further

validated with the BF-score in the Results section. Other popular models like UNet and Deep LabV3+ were also tried in the pipeline, but they all failed to map the boundary in most cases even when with a high IoU score [17,18].

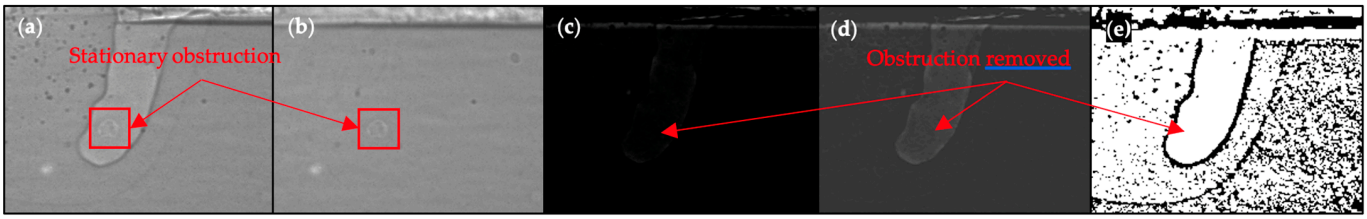


Figure 5. The Subtract step in the filter removes a stationary obstruction affiliated to the dataset: (**a**) a standardized image with the obstruction inside the keyhole; (**b**) the subtraction value as an image, average of 40 previous images in the same dataset, containing the obstruction; (**c**) the subtracted image, where the obstruction is removed by the Subtract step; (**d**) the rescaled subtracted image for visualization; (**e**) binarized image, the output of the filter, with no obstruction.

However, the proposed tool may also encounter segmentation errors, like run 2 testing dataset 3 mentioned in the Results section. Two failed segmentation examples are shown in Figure 6, where in the first example (Figure 6a,b), BASNet predicts no keyhole region, resulting in an IoU and BF-score of less than 0.1. In the second example, shown in Figure 6c,d, BASNet failed to predict the "tail" of the keyhole region. As shown in Figure 6e, a small number of similar errors contribute to the relatively low average IoU and BF-score, while in most cases, BASNet can accurately label the keyhole region. These errors could be attributed to the small keyhole size of dataset 3, which leads to inconsistency in the prediction area with other datasets. Consistency can potentially be achieved by adding a cropping module in the filter or introducing more training data with similar sizes to reduce the effect of keyhole size on the prediction of BASNet. In addition, another factor behind segmentation errors is the fuzzy imaging background for dataset 3, which results in unclear keyhole region contour of the filtered images. Further fine-tuning of the filter parameter N could potentially improve the clarity of the image with a distinct keyhole boundary.

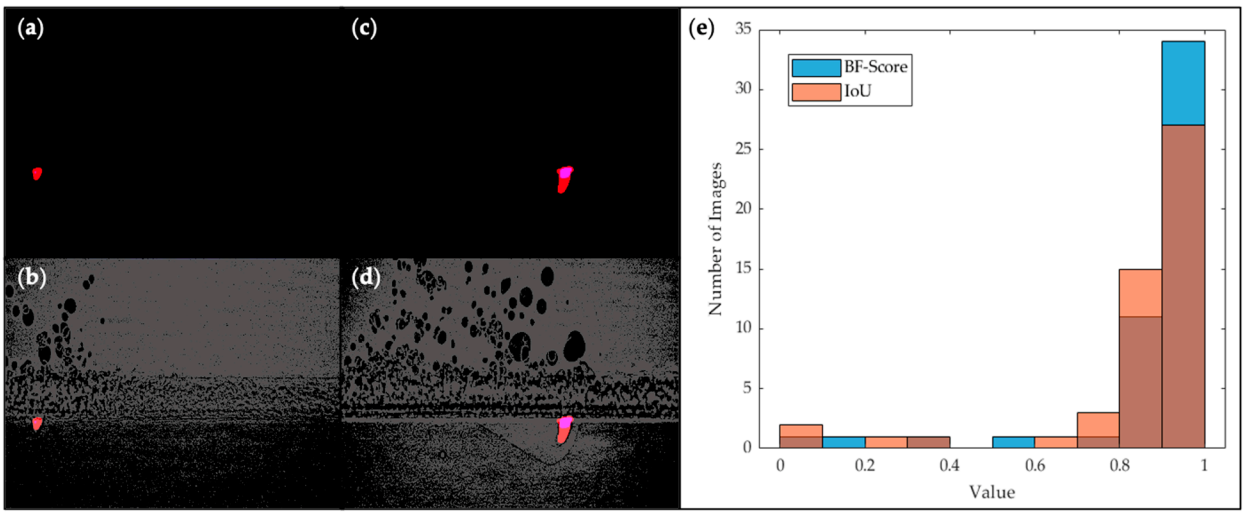


Figure 6. Segmentation errors in run 2, testing dataset 3: (**a**) overlay of the predicted label and ground truth of a failed segmentation example, no label is predicted by BASNet (predicted label in blue, ground truth in red, intersection in purple); (**b**) overlay on the input filtered image; (**c**) another failed segmentation example, where the "tail" of the keyhole region is missed by prediction; (**d**) overlay on the input filtered image; (**e**) histogram of the distribution of tested images in testing dataset 3 in run 2 regarding two evaluation matrices (Number of images with respect to BF-Score in blue, IoU in red, and overlapped in darker orange). Blue area in (**a**–**d**) is barely visible because there is little predicted area that is not ground truth.

So far, the current algorithm cannot process X-ray images with significant differences in experiment setups, and re-training might be needed for prospective users to implement the algorithm. In the future, more training images will be labeled to further optimize this algorithm and to enhance the versatility and robustness of the model.

5. Conclusions

In this study, a deep-learning-based segmentation tool that is capable of automatic segmentation of keyhole morphology in X-ray images with a filter and a trained network pipeline was developed. This tool is validated, with an average IoU of 0.9124 and an average BF-score of 0.9281 on X-ray images from different experiments, proving its high accuracy both in area and boundary, with cross-validation of 300 training and 100 testing images/labeling for each trail.

This work illustrates a repeatable approach for prospective users to automatically generate massive keyhole morphology data with high accuracy on area and boundary from X-ray images. Sufficient morphology data will support developing data-driven analysis of LPBF processes to further improve the quality of additively manufactured products.

Supplementary Materials: The following supporting information can be downloaded at: https://github.com/WilliamDongSH/KeyholeSeg, Computer Source Code S1: MATLab filter for keyhole segmentation; Computer Source Code S2: MATLab checker for keyhole segmentation accuracy performance; Computer Source Code S3: Modified BASNet code with loss visualization and logging; Computer Source Code S4: Trained BASNet model.

Author Contributions: Conceptualization, L.C. and D.M.; methodology, W.D., J.L., C.Y., Y.Z., S.K. and D.M.; software, W.D., J.L., C.Y., Y.Z. and S.K.; validation, W.D.; formal analysis, W.D., J.L., C.Y., Y.Z., S.K. and D.M.; investigation, W.D. and D.M.; resources, W.D., L.C. and D.M.; data curation, W.D., Q.G.; writing—original draft preparation, W.D., L.C. and D.M.; writing—review and editing, W.D., L.C. and D.M.; visualization, W.D.; supervision, L.C. and D.M.; project administration, Q.G., D.M., L.C.; funding acquisition, W.D. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the U.S. National Science Foundation grant numbers [2002840 and 2011354] except for D.M., who was supported by U.S. National Science Foundation Training-based Workforce Development for Advanced Cyberinfrastructure (CyberTraining) award No. 2017072.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The computer source code is publicly available at: https://github.com/WilliamDongSH/KeyholeSeg. Data used in this study is available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank Ryan Jacobs and Yingwei Son for their technical help in this study. This research used resources of the Advanced Photon Source, a US Department of Energy (DOE) Office of Science user facility operated for the DOE Office of Science by Argonne National Laboratory under contract DE-AC02-06CH11357.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chowdhury, S.; Yadaiah, N.; Prakash, C.; Ramakrishna, S.; Dixit, S.; Gupta, L.R.; Buddhi, D. Laser Powder Bed Fusion: A State-of-the-Art Review of the Technology, Materials, Properties & Defects, and Numerical Modelling. *J. Mater. Res. Technol.* **2022**, *20*, 2109–2172. [CrossRef]
2. Abd-Elaziem, W.; Elkatatny, S.; Abd-Elaziem, A.-E.; Khedr, M.; Abd El-baky, M.A.; Hassan, M.A.; Abu-Okail, M.; Mohammed, M.; Järvenpää, A.; Allam, T.; et al. On the Current Research Progress of Metallic Materials Fabricated by Laser Powder Bed Fusion Process: A Review. *J. Mater. Res. Technol.* **2022**, *20*, 681–707. [CrossRef]
3. Dev Singh, D.; Mahender, T.; Raji Reddy, A. Powder Bed Fusion Process: A Brief Review. *Mater Today Proc.* **2021**, *46*, 350–355. [CrossRef]

4. Zhao, C.; Parab, N.D.; Li, X.; Fezzaa, K.; Tan, W.; Rollett, A.D.; Sun, T. Critical Instability at Moving Keyhole Tip Generates Porosity in Laser Melting. *Science 1979* **2020**, *370*, 1080–1086. [CrossRef] [PubMed]
5. Martin, A.A.; Calta, N.P.; Hammons, J.A.; Khairallah, S.A.; Nielsen, M.H.; Shuttlesworth, R.M.; Sinclair, N.; Matthews, M.J.; Jeffries, J.R.; Willey, T.M.; et al. Ultrafast Dynamics of Laser-Metal Interactions in Additive Manufacturing Alloys Captured by in Situ X-Ray Imaging. *Mater. Today Adv.* **2019**, *1*, 100002. [CrossRef]
6. Zhang, Y.; Zhang, J. Modeling of Solidification Microstructure Evolution in Laser Powder Bed Fusion Fabricated 316L Stainless Steel Using Combined Computational Fluid Dynamics and Cellular Automata. *Addit. Manuf.* **2019**, *28*, 750–765. [CrossRef]
7. Alphonso, W.E.; Baier, M.; Carmignato, S.; Hattel, J.H.; Bayat, M. On the Possibility of Doing Reduced Order, Thermo-Fluid Modelling of Laser Powder Bed Fusion (L-PBF)—Assessment of the Importance of Recoil Pressure and Surface Tension. *J. Manuf. Process.* **2023**, *94*, 564–577. [CrossRef]
8. Qu, M.; Guo, Q.; Escano, L.I.; Clark, S.J.; Fezzaa, K.; Chen, L. Mitigating Keyhole Pore Formation by Nanoparticles during Laser Powder Bed Fusion Additive Manufacturing. *Addit. Manuf. Lett.* **2022**, *3*, 100068. [CrossRef]
9. Wang, L.; Zhang, Y.; Chia, H.Y.; Yan, W. Mechanism of Keyhole Pore Formation in Metal Additive Manufacturing. *NPJ Comput. Mater.* **2022**, *8*, 22. [CrossRef]
10. Bayat, M.; Thanki, A.; Mohanty, S.; Witvrouw, A.; Yang, S.; Thorborg, J.; Tiedje, N.S.; Hattel, J.H. Keyhole-Induced Porosities in Laser-Based Powder Bed Fusion (L-PBF) of Ti6Al4V: High-Fidelity Modelling and Experimental Validation. *Addit. Manuf.* **2019**, *30*, 100835. [CrossRef]
11. Chen, L.; Yao, X.; Tan, C.; He, W.; Su, J.; Weng, F.; Chew, Y.; Ng, N.P.H.; Moon, S.K. In-Situ Crack and Keyhole Pore Detection in Laser Directed Energy Deposition through Acoustic Signal and Deep Learning. *Addit. Manuf.* **2023**, *69*, 103547. [CrossRef]
12. Ren, Z.; Gao, L.; Clark, S.J.; Fezzaa, K.; Shevchenko, P.; Choi, A.; Everhart, W.; Rollett, A.D.; Chen, L.; Sun, T. Machine Learning–Aided Real-Time Detection of Keyhole Pore Generation in Laser Powder Bed Fusion. *Science 1979* **2023**, *379*, 89–94. [CrossRef] [PubMed]
13. Pyeon, J.; Aroh, J.; Jiang, R.; Verma, A.K.; Gould, B.; Ramlatchan, A.; Fezzaa, K.; Parab, N.; Zhao, C.; Sun, T.; et al. Time-Resolved Geometric Feature Tracking Elucidates Laser-Induced Keyhole Dynamics. *Integr. Mater. Manuf. Innov.* **2021**, *10*, 677–688. [CrossRef]
14. Zhang, J.; Lyu, T.; Hua, Y.; Shen, Z.; Sun, Q.; Rong, Y.; Zou, Y. Image Segmentation for Defect Analysis in Laser Powder Bed Fusion: Deep Data Mining of X-Ray Photography from Recent Literature. *Integr. Mater. Manuf. Innov.* **2022**, *11*, 418–432. [CrossRef]
15. Qin, X.; Fan, D.-P.; Huang, C.; Diagne, C.; Zhang, Z.; Sant'Anna, A.C.; Suàrez, A.; Jägersand, M.; Shao, L. Boundary-Aware Segmentation Network for Mobile and Web Applications. *arXiv* **2021**, arXiv:2101.04704.
16. Csurka, G.; Larlus, D.; Perronnin, F. What Is a Good Evaluation Measure for Semantic Segmentation? In Proceedings of the British Machine Vision Conference, Bristol, UK, 9–13 September 2013; Burghardt, T., Damen, D., Mayol-Cuevas, W., Mirmehdi, M., Eds.; Visual Information Laboriatory (VIL): Bristol, UK, 2013; pp. 32.1–32.11.
17. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 833–851.
18. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.