7-2023

Image Segmentation with Human-in-the-loop in Automated De-caking Process for Powder Bed Additive Manufacturing

Vincent Opare Addo Asare-Manu
The University of Texas Rio Grande Valley

IMAGE SEGMENTATION WITH HUMAN-IN-THE-LOOP IN AUTOMATED DE-CAKING PROCESS FOR POWDER BED ADDITIVE MANUFACTURING

A Thesis

by

VINCENT OPARE ADDO ASARE-MANU

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE IN ENGINEERING

Major Subject: Manufacturing Engineering

The University of Texas Rio Grande Valley

July 2023

IMAGE SEGMENTATION WITH HUMAN-IN-THE-LOOP IN AUTOMATED DE-CAKING PROCESS FOR POWDER BED ADDITIVE MANUFACTURING

A Thesis
by
VINCENT OPARE ADDO ASARE-MANU

COMMITTEE MEMBERS

Zhaohui Geng, Ph.D.
Chair of Committee

Jianzhi Li, Ph.D.
Committee Member

Douglas Timmer, Ph.D.
Committee Member

July 2023

ABSTRACT

Asare-Manu, Vincent O. A., Image Segmentation with Human-in-the-loop in Automated De-caking Process for Powder Bed Additive Manufacturing. Master of Science in Engineering (MSE), July, 2023, 64 pp., 6 tables, 32 figures, references, 69 titles.

Additive manufacturing (AM) becomes a critical technology that increases the speed and flexibility of production and reduces the lead time for high-mix, low-volume manufacturing. One of the major bottlenecks in further increasing its productivity lies around its post-processing procedures. This work focuses on tackling a critical and inevitable step in powder-bed additive manufacturing processes, i.e., powder cleaning or de-caking. Pressing concerns can be raised with human involvement when performing this task manually. Therefore, a robot-driven automatic powder cleaning system could be an alternative to reducing time consumption and increasing safety for AM operators. However, since the color and surface texture of the powder residuals and the sintered parts are similar from a computer vision perspective, it can be challenging for robots to plan their cleaning path. This study proposes a machine learning framework incorporating image segmentation and eye tracking to de-cake the parts printed by a powder bed additive manufacturing process. The proposed framework intends to partially incorporate human biological behaviors to increase the performance of an image segmentation algorithm to assist the path planning for the robot de-caking system. The proposed framework is verified and evaluated by comparing it with the state-of-the-art image segmentation algorithms. Case studies were utilized to validate and

verify the proposed human-in-the-loop algorithms. With a mean accuracy, f1-score, precision, and IoU score of 81.2%, 82.3%, 85.8%, and 66.9%, respectively, the suggested HITL eye tracking plus segmentation framework produced the best performance out of all the algorithms evaluated and compared. Regarding computational time, the suggested HITL framework matches the running times of the other test existing models, with a mean time of 0.510655 seconds and a standard deviation of 0.008387. Finally, future works and directions are presented and discussed. A significant portion of this work can be found in (Asare-Manu et al., 2023).

DEDICATION

This work is dedicated to my family and all my loved ones. God bless you all.

ACKNOWLEDGMENTS

I extend my profound gratitude to Zhaohui Geng, Ph.D., my thesis committee chair, for his unwavering support, supervision, and guidance. Also, I am very grateful to Jianzhi Li, Ph.D., and Douglas Timmer, Ph.D., for their recommendations and direction toward making this work successful. I thank all my lab mates, particularly Sachithra Karunathilake, for your steadfast input, support, and encouragement. This work was supported by the US Department of Defense Manufacturing Engineering Education Program (MEEP) under Award number N00014-19-1-2728. My sincere gratitude also goes to the Greater Brownsville Incentive Cooperation, which supports our Future SQC projects.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The emergence of additive manufacturing (AM) technology creates more opportunities to produce flexible parts with complex features or intricate structures in critical applications, such as aerospace (Tahmina et al., 2023) or biomedicine (Geng & Bidanda, 2020). Unlike the conventional machining processes that remove excess materials to generate geometric features, AM prints a part by adding materials layer-by-layer. In this way, complex features or intricate structures can be formed by the stack of profiles on each layer, providing more design flexibility opportunities. AM allows complex geometries can be created, customizations can be made, quick prototyping can be done, and on-demand production can be done.

There are many types of AM methods, such as Fused Deposition Modeling (FDM) (Mohamed et al., 2015), Stereolithography (SLA) (Huang et al., 2020), and Digital Light Projection (DLP) (Geng & Bidanda, 2022), which are widely used in industries. However, Powder-bed-based AM processes, such as selective laser melting/sintering (SLM/SLS) (Kruth et al., 2005; Yap et al., 2015), binder jetting (Ziaee & Crane, 2019), Direct Metal Laser Sintering (DMLS) (Bertol et al., 2010), Electron Beam Melting (EBM), and multi-jet fusion (Cai et al., 2021), are among the most representative technologies in the AM family. To bind the particles together in a binder jetting process, a coating of powdered material is spread out, and a liquid binding agent is applied only where it is needed.

DMLS AM method is similar to the SLS method, as it can produce metal mold inserts. It uses liquid phase sintering; the procedure uses a laser that is held close to the metal powder. The SLS AM method can process any powdered material, including polymers, hard metals, ceramics, and sand (Kruth et al., 2003). Heat sources, including laser or electron beam and binder, are utilized to bond the selective regions on each layer of the metal or polymer powders on a powder bed based on the original design. The part is printed in layers, and a new layer of powder is spread on top of the preceding layer until the whole print is complete. Once the object is fully printed, it must be cooled until hardened before proceeding to the powder extraction. The powder extraction process should not be done until the part is solidified, as the powder around the object acts as support during printing. Powder bed fusion has benefits, including the capacity to make objects with complicated geometries and high resolution and utilizing various materials. Figures 1-6 show images of all the types of AM processes discussed (Geng & Bidanda, 2022).

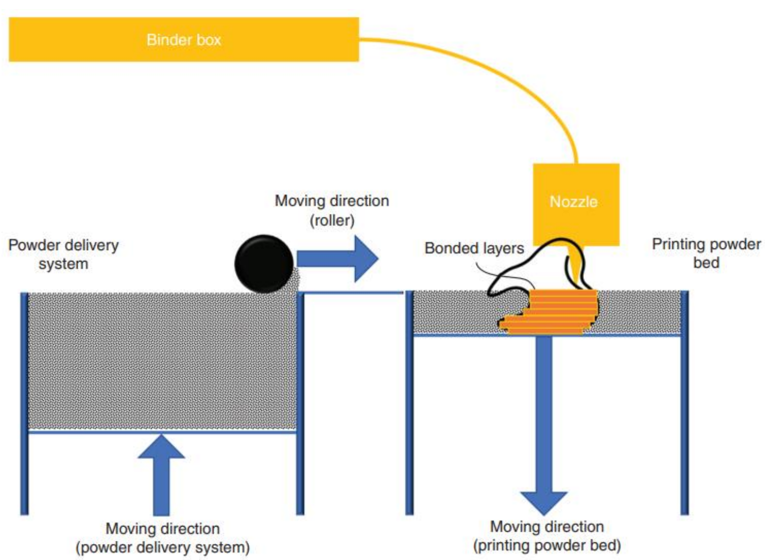


Figure 1: Binder jetting process

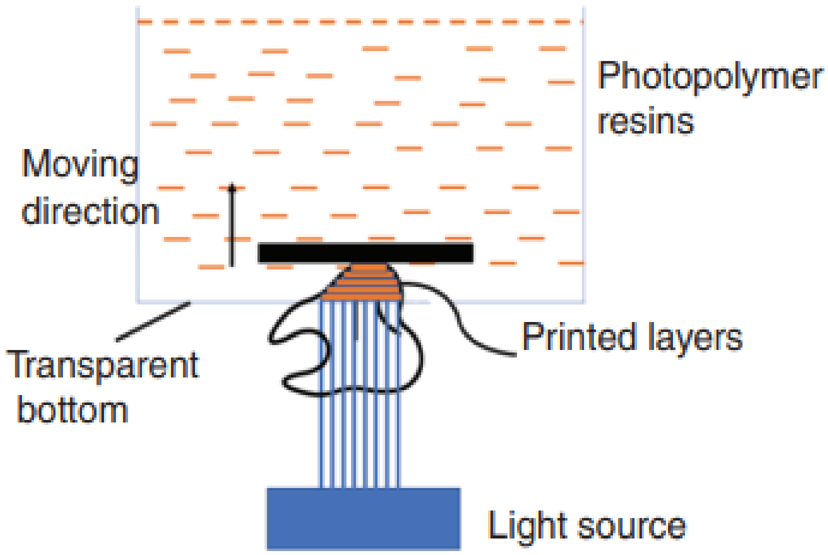


Figure 2: 3D Digital light projection process

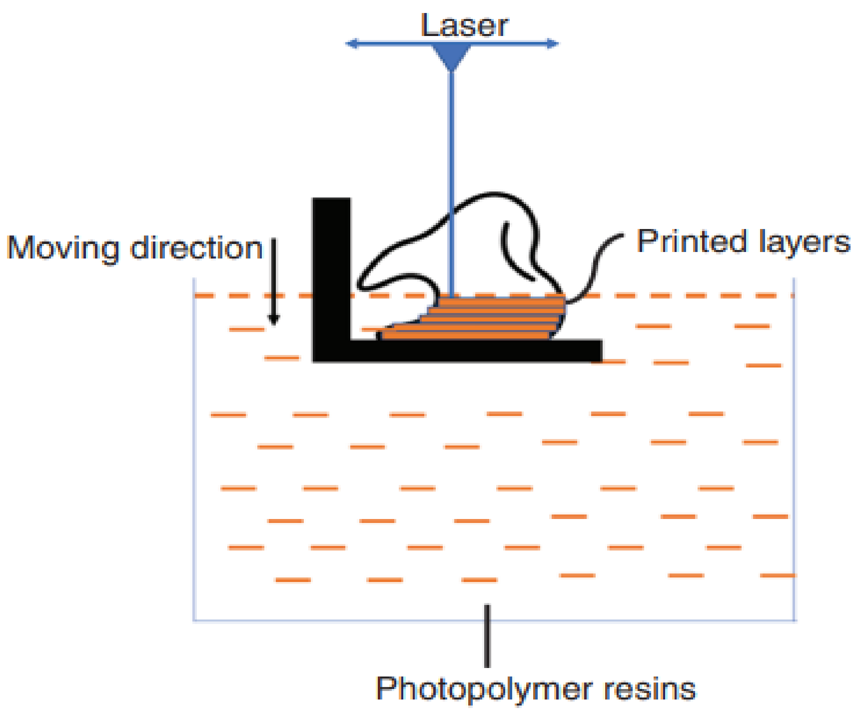


Figure 3: Stereolithography 3D printing process

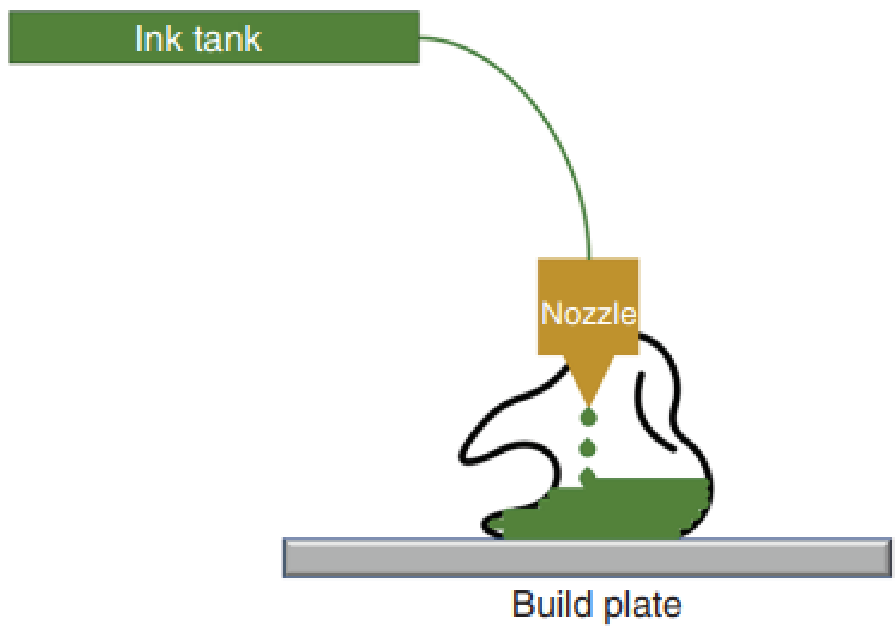


Figure 4: Mechanism of inkjet printing process

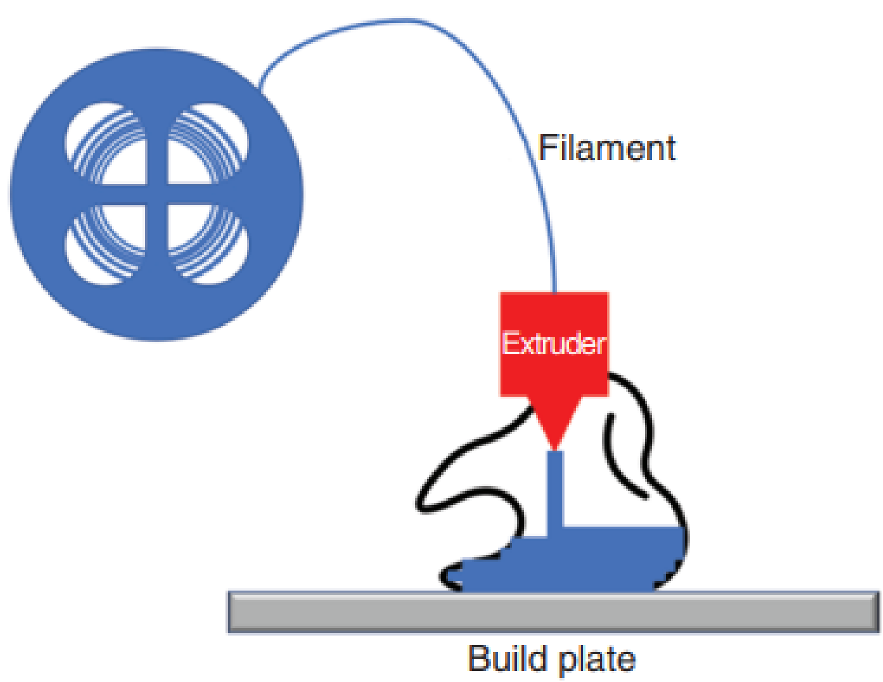


Figure 5: Fused Deposition Modeling 3D printing mechanism

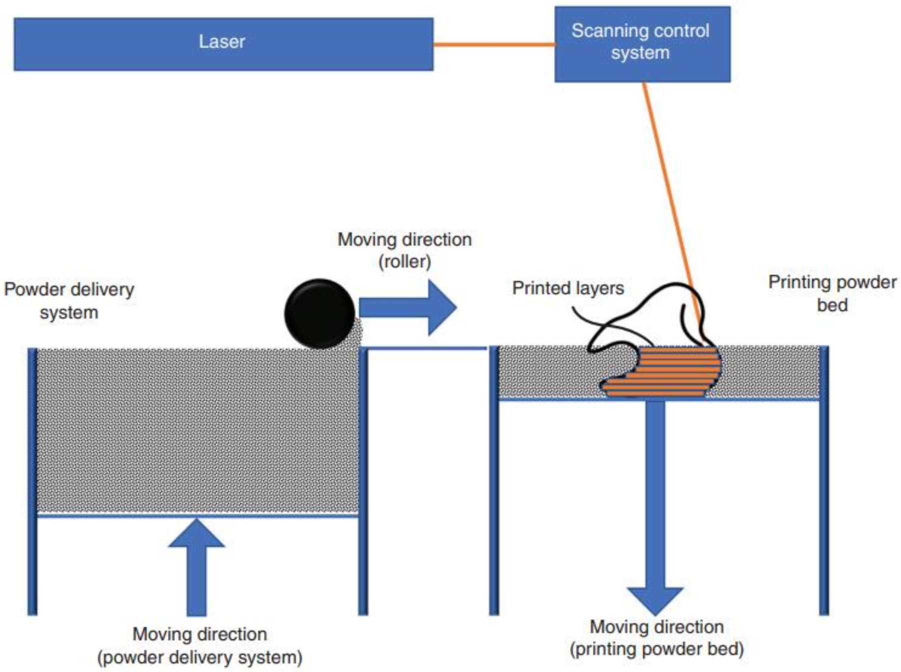


Figure 6: Selective laser sintering process.

On the other hand, necessary post-processing procedures are required in these processes to get the final product. Post-processing methods can generally be grouped into four steps: powder cleaning, removal of supporting structures, removal of part, and surface finishing for a metallic part or polymerization for a product printed by polymer powders (Akbay et al., 2022; Cuellar et al., 2018). This study primarily focuses on powder cleaning - the de-caking process. Its primary objective is to clean and remove the excess powder from the powder bed after printing. The de-caking process is quite straightforward, but it is also inevitable and influences the subsequent post-processing steps and further affects the print quality, surface finishing, and support detaching (Kumbhar & Mulay, 2018; Vayre et al., 2012). Insufficient de-caking of the printed parts could result in undesirable and unacceptable surface quality, un-sintered dust, sags, and unstable junction points, rendering parts porous and low built quality (Akbay et al., 2022).

Several techniques are involved in the de-caking process, such as applying mechanical agitation to break and separating the powder from the printed part. However, the de-caking process is conventionally manual, in which an operator manually utilizes a vacuum cleaner and a brush to clean the powder bed. Since then, it can call for a significant amount of time and effort in this post-processing step and become a bottleneck in increasing the productivity of AM and fostering its wider adoption by the industry. A typical manual de-caking operation is presented in Figure 7, while the powder-bed condition after cleaning for about 45 minutes can be seen in Figure 8. Besides, the time consumption and the quality of the de-caking process also heavily depend on the proficiency of the operators. Thus, the time, effort, and cost involved in manual post-processing can be highly variable. This becomes a major concern in additive manufacturing, especially considering the increasing need for AM in medium-to-large scale production (Cuellar et al., 2018).

Figure 7: Manual de-caking of powder for the SLM process.

Figure 8: Build chamber after powder cleaning for 45 minutes.

Furthermore, when exposed to metallic or nonmetallic powders, powder cleaning can raise critical occupational and safety concerns for the operators (Arrizubieta et al., 2020). Also, powdered feedstocks, especially active ones such as aluminum or magnesium powders, could cause fire hazards with careless operations or inappropriate training (Roth et al., 2019). Fine powders for powder-bed-based AM could potentially create inhalation or dermal exposure, raising allergic or toxic issues (Ohsawa, 2009). Generally, operators must wear masks or protective suits depending on the specific lab policies or operational procedures. Notwithstanding that, the time-consuming de-caking process still raises concerns about occupational safety and human factors.

As an alternative solution, a robotic arm is proposed to replace a human operator to perform the de-caking task. This way, the process is expected to be faster and reduce health concerns because of the reduced exposure between human operators and metallic or plastic powder. Specifically, the solution is that a robot arm carries the vacuum cleaner and the brush to clean the excess powder in the build plate. Utilizing a robot for this application aims to reduce total operational cost, fatigue, and repeatability of the cleaning process. This idea seeks to tackle a significant drawback of the manual post-cleaning process related to the direct human involvement and contact with the metallic powders used for the printing. Irrespective of the standards and safety regulations that guide AM and 3D printing, human health is still being compromised as operators may develop respiratory-related diseases over time due to a long time and accumulated exposure to metallic powders. The proposed system could not only resolve the hazards of human operations but also, with optimized path planning, reduce the time for the de-caking process. To set up this system, many components need to be considered. For example, the build chamber size for the commercialized powder bed AM systems is small, while the door for the robot to enter the chamber can be even smaller. Thus, the collision avoidance algorithm needs to be developed so that the robot arm can work more efficiently in this confined space while interacting with the sintered parts.

Moreover, the path planning for the robot arm is always the holy grail in robotics. Since the "environment" of the build plate, i.e., the volume of the excess powder, changes along the robot operation, the path planning needs to be adaptive to such change for more efficient and effective de-caking. Some other considerations and tasks of a robot-driven de-caking system can be found in (Nguyen et al., 2020). However, the previous work does not consider a practical printing environment, as no powder residuals were located around the part to be cleaned. Also, the authors utilized a 3D camera in a fixed position, which may not be applicable for chambers in powder-bed AM systems with limited spaces. Most importantly, the image processing and cleaning time were not reported, which can be an issue when considering the efficiency in AM.

This study focuses on developing a computer vision (CV) algorithm that enables the abovementioned concerns for the robot de-caking system. CV, as a section of artificial intelligence (AI), trains the system to interpret and understand the visual world for decisionmaking purposes (Voulodimos et al., 2018). Specifically, in this work, the primary objective of the CV system is to differentiate the powder from the sintered parts and the concrete build chamber. An efficient HITL algorithm could guide the robot arm to where the powder residuals are and assist in efficient path planning. However, this task can be challenging for the de-caking process. First, it can be challenging for a CV algorithm to differentiate the powder and non-powder (printed part and surrounding) regions. For powder bed additive manufacturing processes, the surface texture and color of the printed part are similar from the computer's visual perspective. Second, even though the location of the parts on the build plate can be extracted from the printing plan, it can be hard and time-consuming to model the powder flow during the de-caking process. Thus, the current state of the cleaning task can be seen as unknown. Lastly, the parts created by some powder bed AM processes are fragile, such as the green part in the binder jetting process. Prolonged vacuum cleaning time could degrade the quality or even destroy it. Therefore, a more efficient and accurate CV algorithm to segment the powder using the image or video data.

In this study, we adopt a human-in-the-loop (HITL) strategy to enhance the performance of the segmentation algorithm. Human knowledge, represented by captured biological information, is incorporated into the segmentation algorithm to increase its intelligence, reliability, and accuracy to make the CV system more intelligent and efficient (X. Wu et al., 2022). The major framework can be separated into two main steps: 1) a conventional image segmentation algorithm roughly differentiates the powder from the part and the background, i.e., the build chamber; 2) an eye-tracking device captures the location of an operator's eyeballs on the image/video during the powder cleaning process, which can be utilized to differentiate the powder residuals versus all non-powder regions distinctively. The proposed framework is more computationally efficient and fast when compared to previous studies, which is more suitable for robot operations and fast path planning.

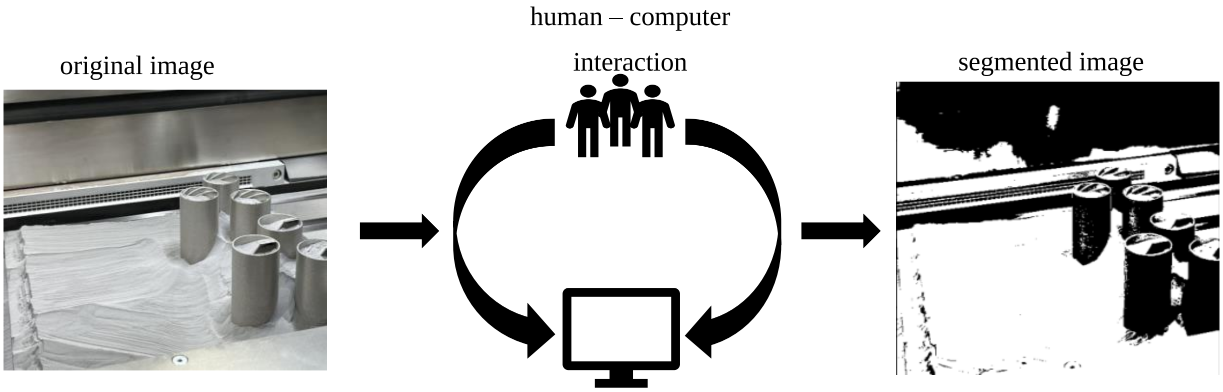


Figure 9: A summary of the Human-in-the-Loop (HITL) process in image segmentation.

The rest of this work is organized as follows. Chapter 2 reviews the classic image segmentation algorithms. In Chapter 3, we present our proposed segmentation framework in detail. Chapter 4 presents the results, discussions, and comparisons between our framework and the other existing segmentation algorithms. Concluding remarks and directions for future work that are motivated by our new framework are presented in Chapter 5.

Research Questions

The research aims to develop a machine learning model incorporating eye tracking for de-caking powder bed additive manufacturing processes. The following two (2) research questions have been presented to set the basis for this research:

1. Can classic existing segmentation algorithms be utilized for powder/part separation?
2. How can human input (eye movement) be automatically incorporated into machine learning algorithms to enhance segmentation performance?

CHAPTER II

LITERATURE REVIEW

Image Segmentation (IS)

As a subset of computer vision (CV), image segmentation partitions images into different regions for further analysis based on some image features (Daniel et al., 2012; Sevak et al., 2018). It is widely used in many applications, such as autonomous driving, manufacturing, and biomedicine (Raut et al., 2009). In the biomedical field, image segmentation algorithms are widely adopted to detect tumors in the human brain and identify cancer cells from the images captured by computed tomography (CT) or magnetic resonance imaging (MRI).

The performance of a segmentation algorithm is significantly dependent on the type, quality (pixel density between the background and foreground), properties or settings (brightness level), image format (color or grayscale) of the image, the effectiveness of the segmentation algorithm, and, more importantly, the training depth or robustness of the model for segmentation (W. Khan, 2014). The image's quality could heavily influence a segmentation algorithm's performance. For example, an image with a distinct pixel density between different clusters and the background could be easier segmented with clear boundaries among different entities. Similarly, properties such as the percentage of brightness, contrast, and shadows, which may not be a challenge for any human operators, could also affect the performance of a segmentation algorithm (Mary Synthuja Jain Preetha et al., 2012).

Many image segmentation algorithms have been developed over the past few decades based on thresholding, edge detection, region-based, and fuzzy theory-based segmentation (W. Khan, 2014), summarized in Figure 4.

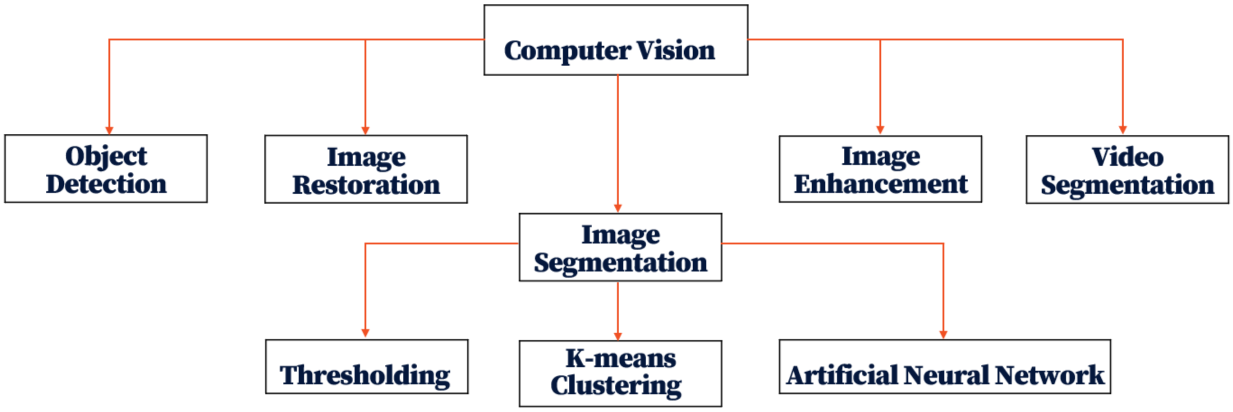


Figure 10: Summary of the discussed Computer Vision (CV) algorithms with a focus on image segmentation

Image Segmentation by Thresholding

There are several ways of thresholding, such as Otsu thresholding (X. Xu et al., 2011), binary thresholding (Bovik, 2009), to-zero thresholding (Bali & Singh, 2015), etc. All thresholding utilizes a grey-scale image and an optimal threshold value (t^*) is either assigned or calculated. Generally, all thresholding methods perform image segmentation by comparing the pixel value at each point in the image (v_i), to the calculated or assigned threshold value (t^*).All the gray level values in the image are set to black (i.e., the intensity, μ_0 = 0) if the pixel value, $v_i \leq t^*$or white (i.e., the intensity, μ_1 =255) if $v_i > t^*$. For binary thresholding where t^* is assigned, a trackbar can be generated to manually obtain the desired point with the target segmented regions (Sahoo et al., 1988). The fast-marching method (A. Xu et al., 2010), Otsu method (X. Xu et al., 2011), entropy criterion and genetic algorithm or Li method (Li & Lee, 1993; K. Wu & Ban, 2011), and particle swarm optimization (Jiang et al., 2012). The calculation of t^* is based on the nature of the objective function of the image segmentation used in the study. According to (Cao et al., 2021), if I is a grayscale image of size $m \times n$, with pixel intensity v_i $(0 \leq v_i \leq 255)$. If we aim to have two classes C_0 and C_1 ,the objective function for Otsu thresholding is given by:

$$\sigma(t) = \sum_{i=0}^{t} p_i \, . (\mu_0(t) - \mu)^2 + \sum_{i=t+1}^{255} p_i \, . (\mu_1(t) - \mu)^2$$

Where μ_0 and μ_1= mean intensities of the two classes

t = threshold value

p_i = probability of a pixel of intensity *i*, which is described as

$$p_i = \frac{f(i)}{m \times n}$$

and *f (i)* represents the number of pixels with intensity *i*

The optimal threshold t* is then calculated by the argmax operation, which yields an argument that gives the maximum threshold value from the objective function $\sigma(t)$. Mathematically, it is then computed by:

$$t^* = \arg\ max\sigma(t)$$

and $$0 \leq t^* \leq 255$$

Given that the mean intensity of the image for the binary classes is then selected following the below:

$$\mu(t) = \begin{cases} \mu_0 = 0, & v_i \leq t^* \\ \mu_1 = 255, & v_i > t^* \end{cases}$$

Edge-based Image Segmentation

Edge-based image segmentation, also called active contouring, is designed to explore the outline or boundaries within an image (W. Khan, 2014). It traces the outline of the objects by detecting discontinuities in brightness. Points of discontinuities are then arranged into line segments or edges (Chakraborty et al., 2017). (Lakshmi et al., 2010) separated the edge detection methods into two major categories, i.e., gray histogram and gradient methods. Interested readers will find relevant research contributions in the reference list (Lakshmi et al., 2010; Wahab et al., 2013; Wesolkowski et al., 2002; Yu et al., 1991; Zaim, 2008). In the edge-based image segmentation method, the objective is to detect edge pixels and then connect them to form the borders between the areas or directly locate the boundaries between the regions.

All edge-based operators are mainly grouped into two – gradient-based and Gaussian based. For any digital image, the gradient-based operators compute the first-order derivative, whiles the Gaussian-based operators compute a second derivative(Narendra & Hareesh, 2011). Examples of edge-based image segmentation methods include active contours and edge linking (Q. Wu & Castleman, 2023), the Canny edge detector, the Prewitt operator, the Marr-Hildreth operator, and the Sobel's method(Kang & Wang, 2007). Like the Prewitt and Robert methods, the Sobel method is classified as a gradient-based operator. On the other hand, the Canny and Marr-Hildreth operator (Laplacian of Gaussian) are Gaussian-based (R. Zhang et al., 2005). Active contours, often known as snakes, can be applied to a picture to match the borders or boundaries of the objects we wish to discover within the image (Hsiao et al., 2005). The contours automatically change their shape to match the selected boundaries. Edge linking connects adjacent edge pixels to form continuous contours and assemble distinct edge segments to form meaningful areas or objects.

For an image pixel of dimensions x and y, and a standard deviation σ, which determines the degree of smoothing and mask size, a Gaussian function is chosen as the low-pass filter, which first smoothens the image with a low-pass filter (Russo & Lazzari, 2005). The above statement is mathematically represented as follows:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

From the equation above, the Laplacian of Gaussian (LoG) or Marr-Hildreth operator is then computed as:

$$LoG = \frac{\partial^2}{\partial x^2} G(x, y) + \frac{\partial^2}{\partial y^2} G(x, y)$$

$$LoG = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

Region-based Image Segmentation

Region-based segmentation techniques are used to explore the segmented regions directly from an image. Image pixels are grouped into various clusters based on their similarities in intensity, texture, and color. Similar to clustering algorithms in unsupervised learning, a homogeneous region is established within the image based on the location of the values of the pixels in the feature space. For example, as a popular clustering algorithm, mean-shift clustering is widely used for segmentation, where the features, such as location, texture, and color, are extracted for every pixel in the image and then classified based on their similar traits (Zhou et al., 2008). (Karoui et al., 2007) utilized texture and level set methods to propose a new way of segmenting an image, which includes a feature selection stage that fits each feature’s weight.

Other segmentation algorithms, such as the ones based on neural networks and deep learning techniques, are also proposed, which are omitted in this paper. Interested readers could refer to the reference list (Kazemi et al., 2008; Terry & Vu, 1993; X. Zhang & Tay, 2007). This class of methods calls for much more computational resources and larger training set for near-human or even superhuman performance.

K-means Clustering

K-means is one of the most straightforward unsupervised learning techniques to handle well-known clustering problems. K-means clustering divides n data points into k clusters to group together comparable data points. K-means clustering can assist in locating these segments by assembling pixels with similar colors or other feature values. In K-means clustering, each pixel is assigned to the cluster with the closest centroid using an iterative process (Shukla, 2014).

The objective function *J* for the K-means algorithm, which minimizes the sum of squared distances between all points and the center of the cluster, is given by:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - c_j||^2$$

Where:

k is the number of clusters

n is the number of data points (pixels)

x_i is the set of each pixel; $\{x_1, x_2, x_3, \ldots\ldots, x_n\}$

c_j is the set of the centroid of each cluster; $\{c_1, c_2, c_3, \ldots\ldots, c_k\}$

$||x_i - c_j||^2$ is the Euclidean distance between each data point and its center

According to (Salihah Abdul-Nasir et al., 2013), the algorithm for K-means clustering for image segmentation consists of the following steps:

1) Initialize the center and the number of *k* clusters.

2) Find the Euclidean distance for each pixel in a picture.

3) Assign each pixel to the closest center based on the distance.

4) Calculate the new position of the centroid once all the pixels have been assigned.

5) Repeat the procedure until the tolerance or error value is met.

6) Resize the pixels in the image cluster.

Artificial Neural Network Based Image Segmentation

This type of segmentation utilizes a machine learning approach, specifically Artificial Neural Networks (ANN), in determining an image segment. Each pixel in an image is matched to a neuron; therefore, the whole image forms a neural network. Next, this neural network, considered as the data, is programmed or trained using any suitable software programming language (Terry & Vu, 1993). The neurons with similar traits are then classified as segments after the training. Specifically for image segmentation, MLP, FFNN, BPNN, and PCNN are popular and most-used algorithms (Van Der Zwaag et al., 2002). In terms of literature and past research work, (Kazemi et al., 2008) and (X. Zhang & Tay, 2007) have proposed fast new image segmentation methods, including the training of a Fuzzy Hopfield Neural Network (HNN) and Fast learning Artificial Neural Network (FLANN).

Like a human brain, the neural network has several layers of neurons that receive inputs processed to produce an output (Abiodun et al., 2018). In image segmentation, the input layer receives the image or the image features. The middle layers perform the computations and transformation of information to predict a meaningful output (Saravanan et al., 2014). The network performs backpropagation to understand the data set further and improve the output with reduced errors.

CHAPTER III

METHODOLOGY

In this study, we aim to develop an image segmentation algorithm for the de-caking process for powder-bed AM, with major concerns regarding image processing speed and performance. Human-in-the-loop (HITL) is defined as a subsection of artificial intelligence (AI) that combines human and machine intelligence to develop or create machine learning algorithms (Chai & Li, 2020; X. Wu et al., 2022). Including prior knowledge from humans in the learning framework makes the model more effective and flexible as the learner does not rely solely on information from limited data (Mosqueira-Rey et al., 2022).

Many researchers describe data pre-processing as part of the human-in-the-loop pipeline as one of the most critical steps. (Chai & Li, 2020) reported that about 80% of the time spent on model development is consumed during pre-processing data stage. It comprises four main subsections: data extraction, cleaning, integration, and dimensional reduction. Through this crucial stage, noisy data or outliers are removed to increase the model performance, and the training set is refined, leading to more representative results that increase the prediction performance (Sevak et al., 2018). Figure 5 simplifies the subgroups of data pre-processing in the HITL pipeline.

When adopting the typical classification algorithm for image segmentation in the de-caking process, the contour of the powder and the printed parts need to be manually drawn using a mouse or a writing pad. This process involves the mechanical movement of hands and human-computer interaction, which can take a long time to get an extensive training set. During this process, the human-computer interaction starts from the human eyeball movement on the screen, which guides the movement of the muscles in the arm. The eyeball would stare at a specific region triggered by the operator's intention, whose movement is much faster than the mechanical movement of the arm. Therefore, if the position of the eyeball can be captured, this locational information could enhance the performance of the segmentation algorithm.

Data Collection and Pre-processing

Initially, the images of the printed parts from an EOS machine Located at the CAMICS laboratory at the University of Texas Rio Grande Valley were taken as sample photos for the experiment. The sample comprised 40 images of 3 random printed parts captured in multiple angles. Out of 40 images, six (6) images were selected based on how different they looked from each other in different angles to proceed with the experiment. Moreover, images from the internet were also included in selected 30 images to test the versatility of the performance of the HITL eye-tracking plus segmentation algorithm proposed in the study.

Prior to data processing, the images were pre-processed by numbering, resizing (500x500 pixels), and formatting (i.e., jpg, jpeg, and png), which was performed by Python programming. For identification, each image was given a number between 1 and 30 and saved in a specified folder.

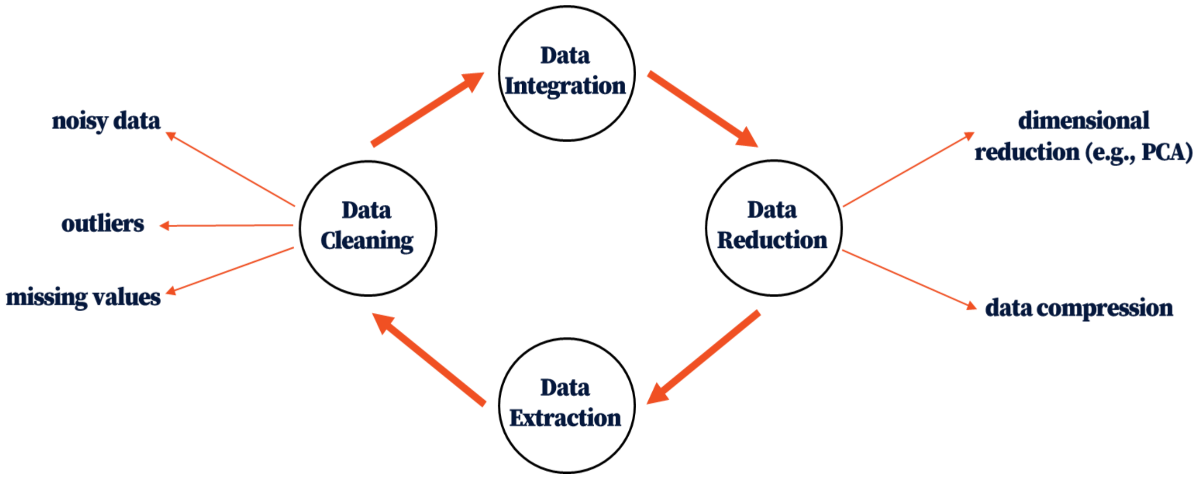


Figure 11: A summary of general data pre-processing procedure in machine learning.

Eye Tracking Software

All 30 images were pooled into six groups of different studies within a single experiment to be presented to the participant to mitigate fatigue. The experimental procedure involved displaying images in two experiments for a specific display period per image for each of the six studies created in the iMotions software. This also implies that six images were fed as the stimulus per study. For example, each experiment comprises study 1 with six images (images 1 to 5) to be tracked by the observer's eye within 10s or 30s per image based on which one of the two experiments described above. In summary, a complete study per experiment for the 30 seconds gaze per image lasts 2 minutes and 30 seconds, whiles that of the 10 seconds gaze per image lasts 50 seconds. The presentation period for each image was prolonged to 30 seconds in the second experiment, compared to just 10 seconds in the first. The purpose of having two separate experiments is to have an idea of the optimal time for the eye-tracker to produce the best results from the heatmaps generated. During the given time, the participant gazes at the powdered areas. After the study, the pool of images was extracted from the software and analyzed. The analysis from the results of the eye-tracking leads to the generation of the heatmaps around the powdered region. All the extracted heatmap images were resized again to 500x500 pixels to facilitate efficient programming, testing, and development of each algorithm to be examined.

iMotions eye-tracking software has been utilized in conjunction with the eye tracker. A vertical alignment with the participant's head was maintained throughout the process by positioning the eye tracker precisely in the middle of the lower bezel of the screen. Following calibration, the eye tracker was tested to ensure its accuracy and stability and eliminate any potential obstructions that could interfere with its operation. The software interface requires the experimenter to input specific display features, such as the screen resolution, to enable the eye tracker to capture the designated screen area accurately.

Thus, the proposed framework incorporates the extra information provided by the location of the operator's eye in the image segmentation. A summary of the proposed algorithm is shown in Figure 14 below. Specifically, we utilize an eye-tracker (Figure 12) to capture the position and time of the eyeball when the operator gets the signal of gazing only at the powder region. The eye-tracker would create a focus region and generate a heatmap around that area of interest, as shown in Figure 7. With the aid of the iMotions software, the human eye is utilized through an eye tracker to generate a segment by forming a heatmap exactly on the focus areas of the eye.

Figure 12: A Smart Eye Aurora peripheral for the iMotions software

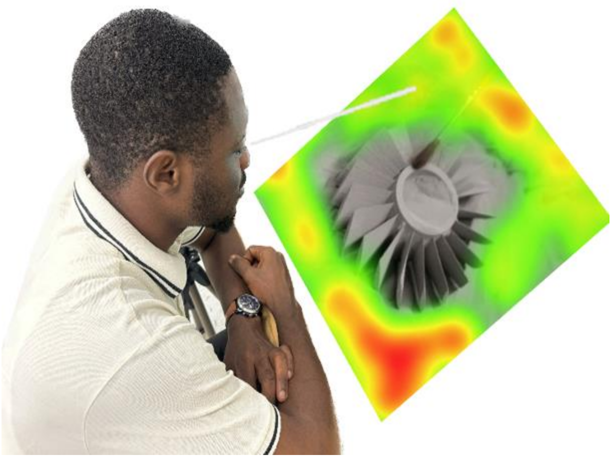


Figure 13: Experimenter focusing on the powder.

Proposed HITL Eye Tracking plus Segmentation Algorithm

The procedure for developing the HITL eye-tracking plus segmentation algorithm, presented in Figure 8, is listed below as follows:

1. An operator focuses on the powder region, and the eye tracker captures the location and time of the operator’s gaze on the screen.
2. The location of the operator’s gaze is presented on a heatmap, whose area and color represent the location, respectively.

3. A K-means thresholding algorithm with 3 clusters (k = 3) roughly segments the heatmap images generated by human supervision from the eye tracker. This is where the human-in-the-loop is combined with the machine learning algorithm, in this case, a k-means clustering algorithm.
4. The segmented images from Step 3 are subjected to a masking technique that further groups the clusters from k=3 to a binary gray image showing black and white colors only. This is achieved through Python programming by converting all images into the HSV format and masking out a specific range of values of pixels that fall with the HSV scale to white.
5. This further generates a properly segmented binary segmented image where the focus region characterized by powder is set to a pixel value of 255 (white), while pixel values falling out of the HSV range (non-powder regions) are set to 0 (black).

The heatmap from the eye-tracking data provides a partially segmented image for the powdered area in the pre-segmented image while including the k-means clustering algorithm to finetune the already segmented image from eye-tracking. The heatmaps from only the eye-tracking experiment cannot be utilized for segmentation over the HITL segmentation plus eye-tracking framework because controlling eyeball movement is relatively difficult and may be unreliable. Instead, the biological information is incorporated into the segmentation algorithm in a weakly informative fashion, which assists in image segmentation.

Based on the proposed model, a robotic arm equipped with a vacuum suction end effector and a vision system such as a CMOS camera can be trained to clean off powder by locating and sucking off the powder regions leaving only the solid or printed part.

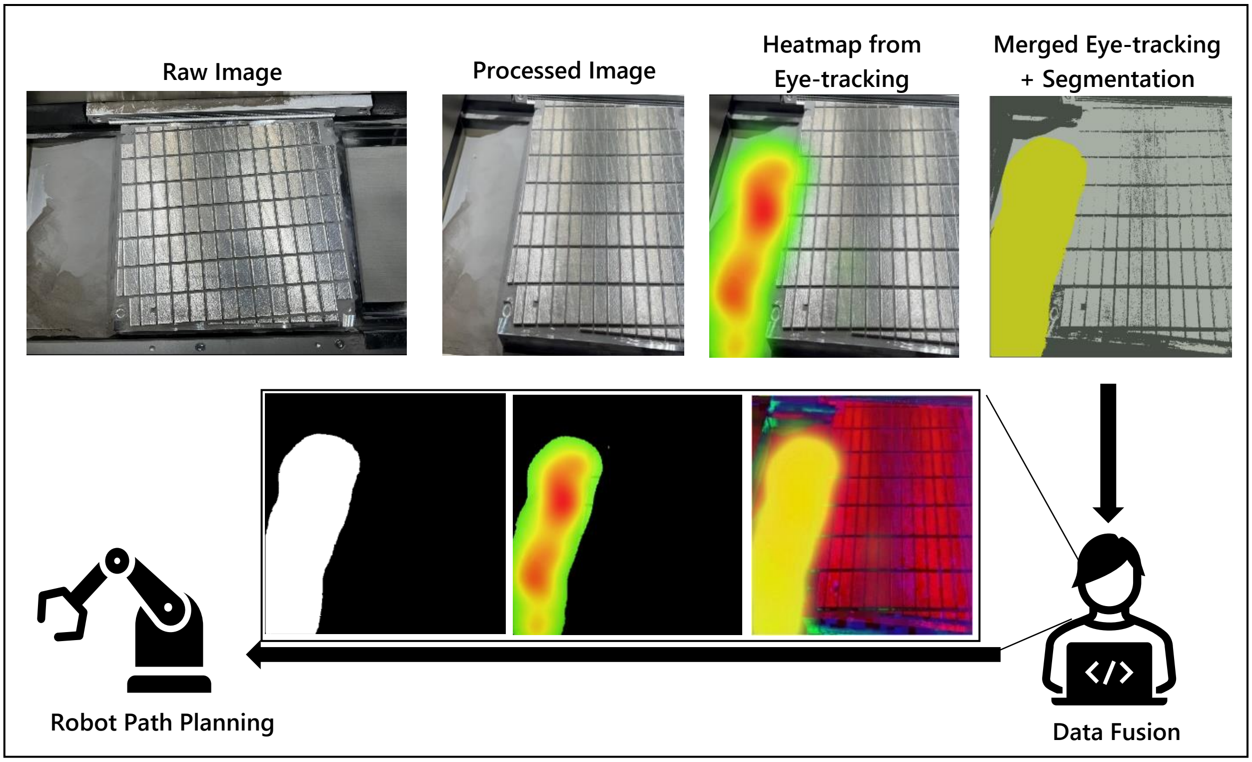


Figure 14: Summary of the proposed HITL eye tracking plus segmentation algorithm.

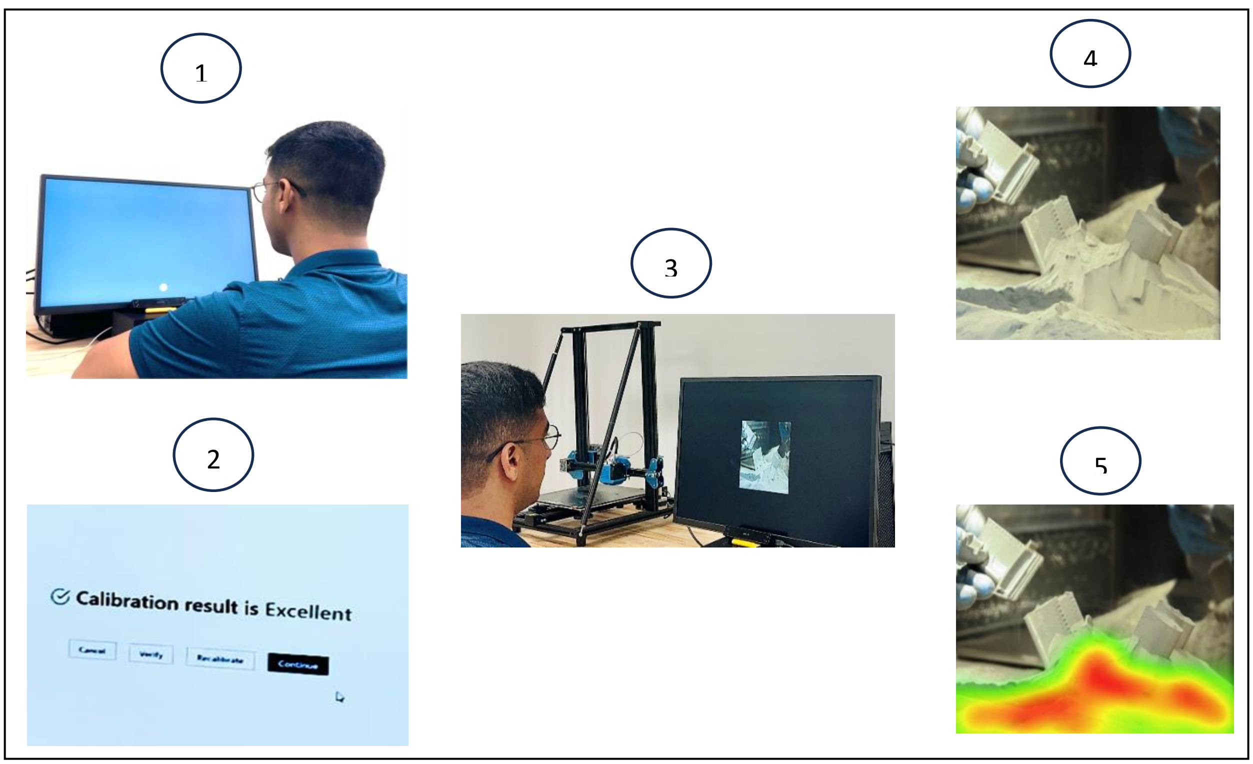


Figure 15: Summary of eye-tracking masking without classic segmentation algorithm.

Technical Specification of the Apparatus

The observer's eye movement was recorded using the iMotions Aurora Smart Eye device through the eyeball movement. This technique uses cutting-edge technology to capture the movement of the eyeball by examining the observer's black pupil and corneal reflection. A Dell Precision 3650 Tower workstation with an Intel Core i9-11900K CPU and 8 cores running at 3.5 GHz was used for the eye-tracking procedure.

The technical characteristics of the eye tracker used in the experiment were as follows: sample rates of 60, 120, and 250 Hz, enabling precise and accurate data collection. The eye-tracking technology requires a 50-to-80-centimeter ideal operating distance between the subject and the eye tracker for the best results. The eye tracker's estimated precision of 0.3 degrees allowed for accurate eye movement measurements. Additionally, the device has a precision of 0.2 degrees, guaranteeing great accuracy.

The system used a time stamp precision of 1 microsecond to ensure temporal precision, providing precise data synchronization. It was observed that the eye tracker also showed reduced latency, enabling real-time observation of eye movement. For the 60 Hz, 120 Hz, and 250 Hz sampling rates, the reported latency was 25 milliseconds, 17 milliseconds, and 12 milliseconds, respectively.

CHAPTER IV

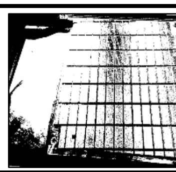
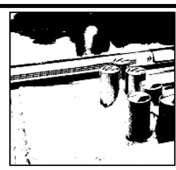
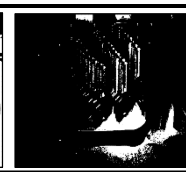
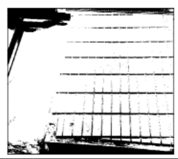
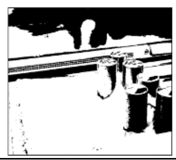
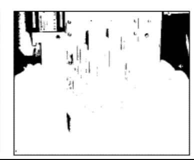
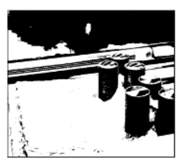
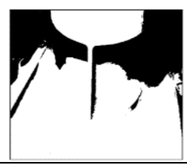
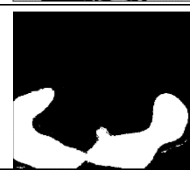
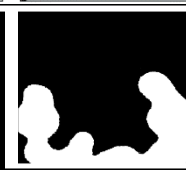
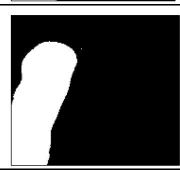
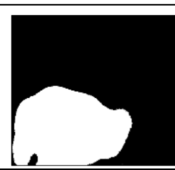
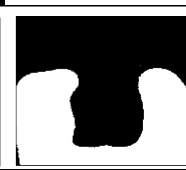
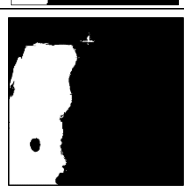
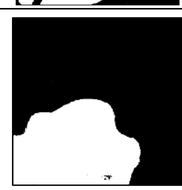
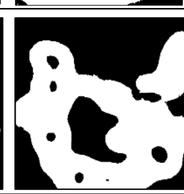
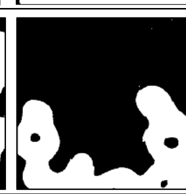
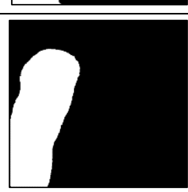
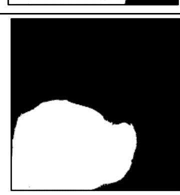
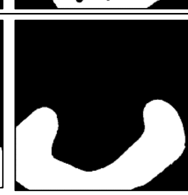
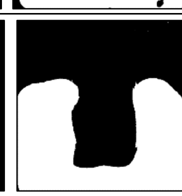
RESULTS AND DISCUSSION

As mentioned above in the introduction, the proposed HITL eye tracking plus segmentation algorithm will be tested and compared to existing segmentation models to test for its performance and computational time.

To evaluate the efficiency of the proposed HITL framework, a sample of 30 images of the de-caking process are gathered from laboratory and internet printing experiments. The variety in the image source establishes the model's adaptability irrespective of the image source or printing environment. Other classic image segmentation algorithms are selected, including Otsu thresholding (Dutta et al., 2022; X. Xu et al., 2011), Li thresholding (Li & Lee, 1993), and K-means clustering. All the algorithms are tested using Python programming with the aid of libraries such as OpenCV (Xie & Lu, 2013), Matplotlib (Hunter, 2007), and scikit-image (Van Der Walt et al., 2014). Five images were randomly selected to display the results from all models to be compared, and the results for each algorithm are presented in Table 1. The results from the remaining 25 images are shown in the appendix.

This section discusses the results of the case study described in Chapter 4. Five images are selected randomly selected to present for comparison and processed by each algorithm.

Table 1: Summary of results from all five sampled images tested by each algorithm.

	1	2	6	13	30
					
Binary Thresholding					
Li Thresholding					
Otsu Thresholding					
Eye tracking Masking (10s)					
Eye tracking Masking (30s)					
HITL Eye tracking + K-means Segmentation (10s)					
HITL Eye tracking + K-means Segmentation (30s)					

In terms of the performance of each algorithm presented in Table 1, even without looking at any of the values for the metrics computed for this study, it can be easily deduced that the top 3 models in no order are HITL eye-tracking plus segmentation (10s), eye-tracking masking only (30s), and HITL eye-tracking plus segmentation (30s). Figure 10 presents a side-by-side comparison of the top 3 models relative to their corresponding ground truth images.

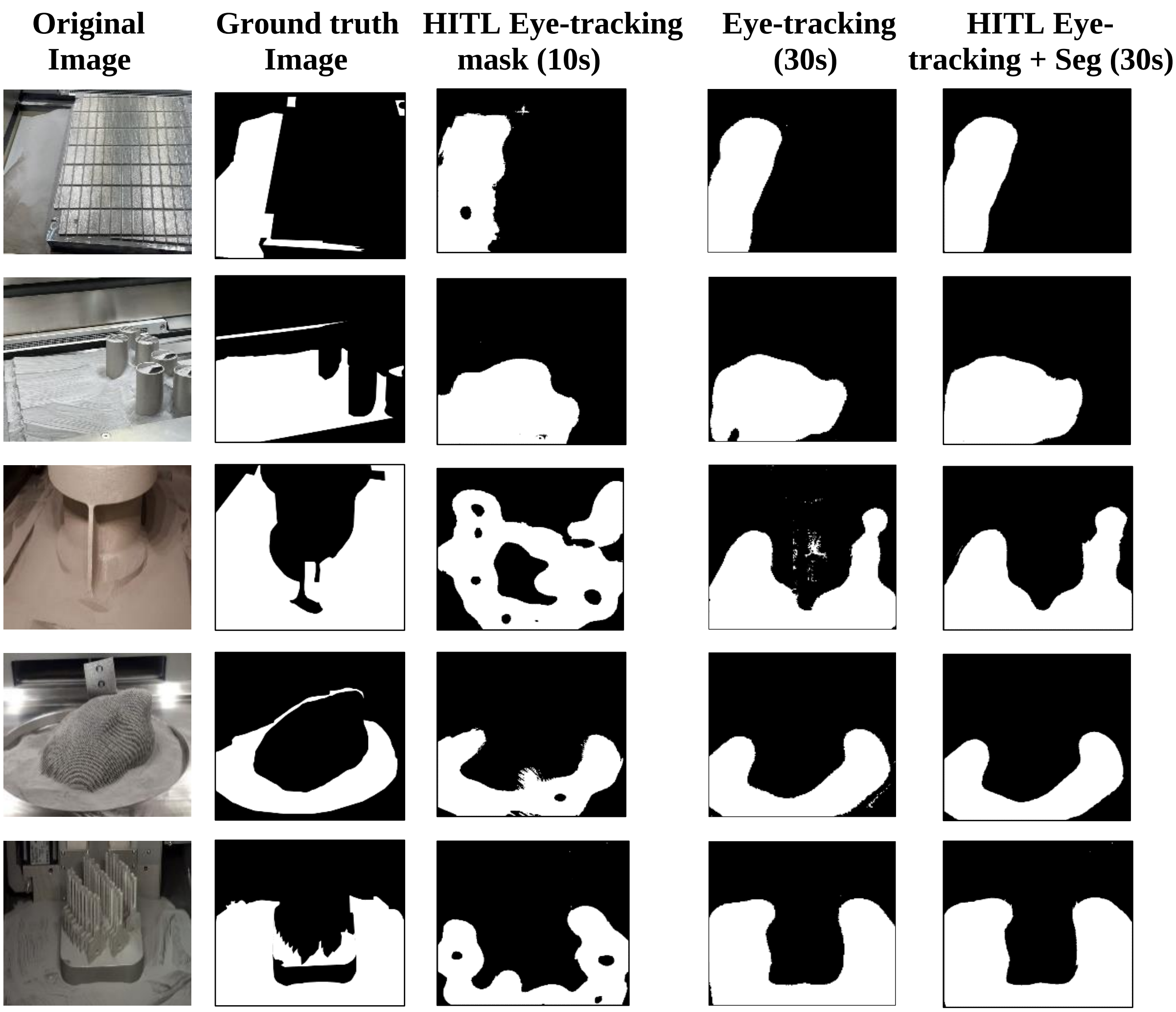


Figure 16: Comparison of top 3 performing algorithms (HITL eye tracking plus segmentation (10s), eye tracking masking only (30s), and HITL eye tracking plus segmentation (30s)) with their respective ground truth images.

Model Performance Evaluation

Based on our aim of executing a binary segmentation and the type of image dataset we have, the classification or model evaluation metrics considered were accuracy (weighted recall), precision, f1-score, and intersection over union (IoU) (Taha & Hanbury, 2015). All these metrics are characterized by four components, namely false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) (Kurmi & Chaurasia, 2018). There are only two classes for a binary segmented image – white (255) and black (0). For this study, the primary focus region is the powder (white segment), which corresponds to the positives, while the non-powder segments (black) are to the negatives. Say white is positive and black is negative; the components of each metric are defined below.

1. True Positive means the model correctly predicted the region as white (powder).
2. False Positive means the model incorrectly predicted the region as white (powder).
3. False Negative means the model incorrectly predicted the region as black (non-powder).
4. True Negative means the model correctly predicted the region as black (non-powder).

It is also important to note that for this study. However, both type I and II errors impact the results of a model. Type I (predicting the powder as a non-powder) error is more substantial and cannot be ignored relative to type II error. Type I errors impact the model performance, which cannot be compromised compared to computational time. In summary, every false prediction affects a model's performance and cleaning time, but depending on the study objectives, performance may be slightly weighted than cleaning time. Type II errors may only result in longer cleaning time with consistent or constant performance since parts and surroundings classified as non-powder are predicted as powder. Therefore, more time is needed than expected also to clean the incorrectly predicted class, the powder. Also, the description of the false prediction, whether FP or FN, leads to type I and II errors respectively.

Pixel Accuracy

Pixel accuracy is the proportion of correctly detected pixels in the predicted segmentation relative to the actual segmentation (T. Rahman et al., 2020). In this work, it was observed from the results obtained that the pixel accuracy is equal to the weighted recall. Although pixel accuracy provides a simple indicator of overall correctness, it may not be suitable for imbalanced datasets where most pixels are in the background class (Wang et al., 2020). Accuracy is mathematically presented as:

$$\text{Pixel Accuracy} = \frac{(\mathrm{TP} + \mathrm{TN})}{\big((\mathrm{TP} + \mathrm{FN}) + (\mathrm{FP} + \mathrm{TN})\big)}$$

Precision

Precision assesses the proportion of foreground pixels that are correctly predicted out of all predicted foreground pixels (H. Zhang et al., 2008). Precision is mathematically presented as:

$$\text{Precision} = \frac{(\mathrm{TP})}{(\mathrm{TP} + \mathrm{FP})}$$

F1-Score / Dice Coefficient

The dice coefficient, also known as the f1-score, is a statistic that assesses the degree of similarity between expected and actual segmentation. It is calculated by dividing twice the intersection by all segmentations representing the expected and actual data. Like all other indicators, the Dice coefficient ranges from 0 to 1, with higher values indicating better performance (Taha & Hanbury, 2015). By mathematical formula, f1-score is represented as:

$$\text{Dice Coefficient} = \frac{(2 \times \mathrm{TP})}{(2 \times \mathrm{TP} + \mathrm{FN} + \mathrm{FP})}$$

Intersection over Union (IoU, Jaccard Index)

The Jaccard Index, often referred to as intersection over union (IoU), determines how much of the segmentation's predicted and actual overlap (M. A. Rahman & Wang, 2016). With respect to image segmentation, the formula below shows the calculation of the Jaccard Index:

$$\text{IoU, Jaccard Index} = \frac{(\text{TP})}{(\text{TP} + \text{FN} + \text{FP})}$$

Results

All 30 photos were evaluated for the criteria specified above, and density plots were generated using Python programming. A density plot, also known as a kernel density plot or kernel density estimate (KDE), is a graphical representation of the probability density function underlying a continuous random variable (Chhabra et al., 2017). It enables visualization of the data distribution and offers a smooth estimate of the density of the data points (Kwon et al., 2020). Through density plots, the patterns and locations of the distributions of all the models can be visually compared by superimposing different density plots on the same graph. Figures 11-14 show the density plots from evaluating all 30 images. On each one of the density plots shown below, the individual tested models are indicated by different colors.

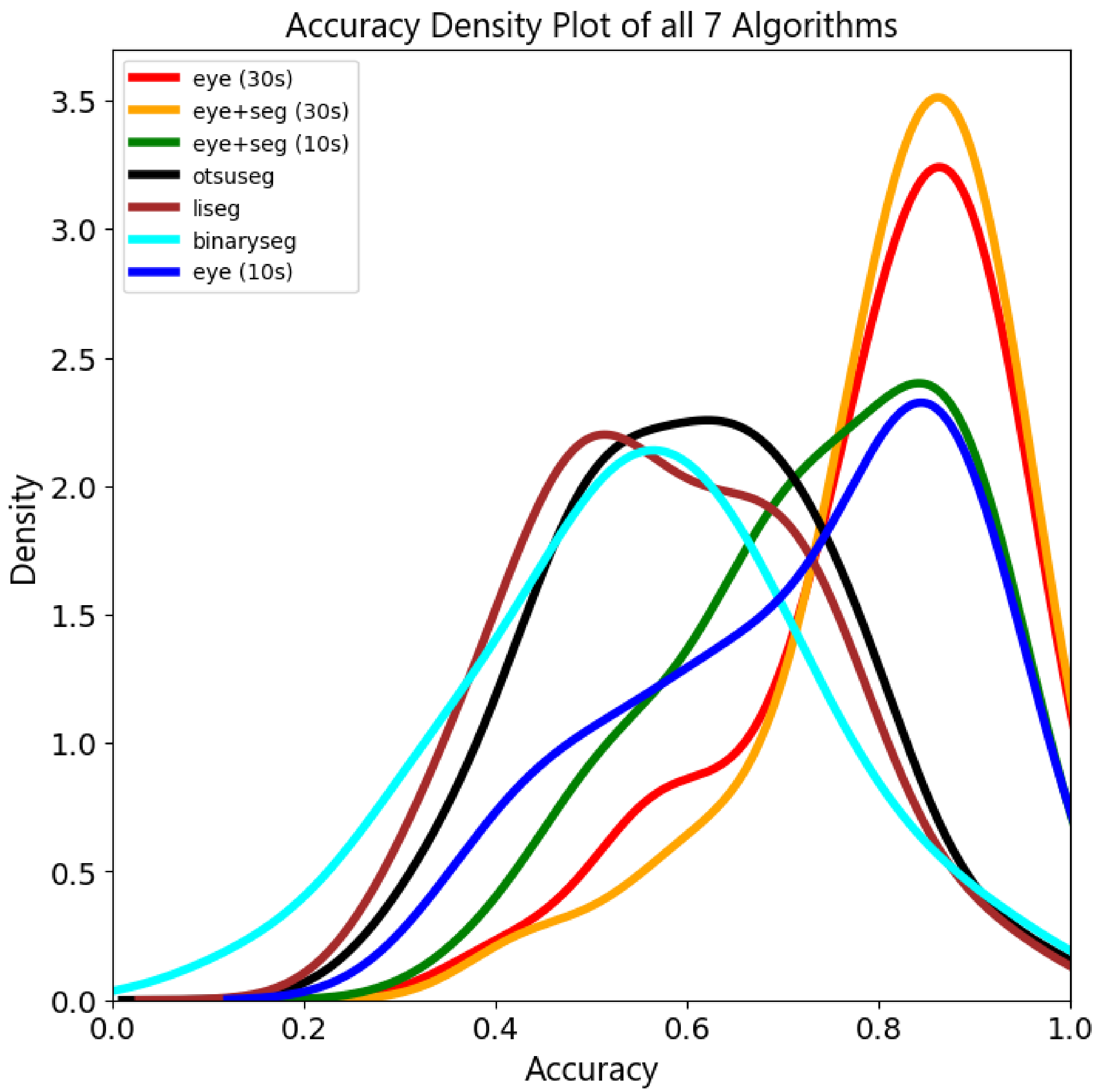


Figure 17: Density plot of the accuracy of all 30 images

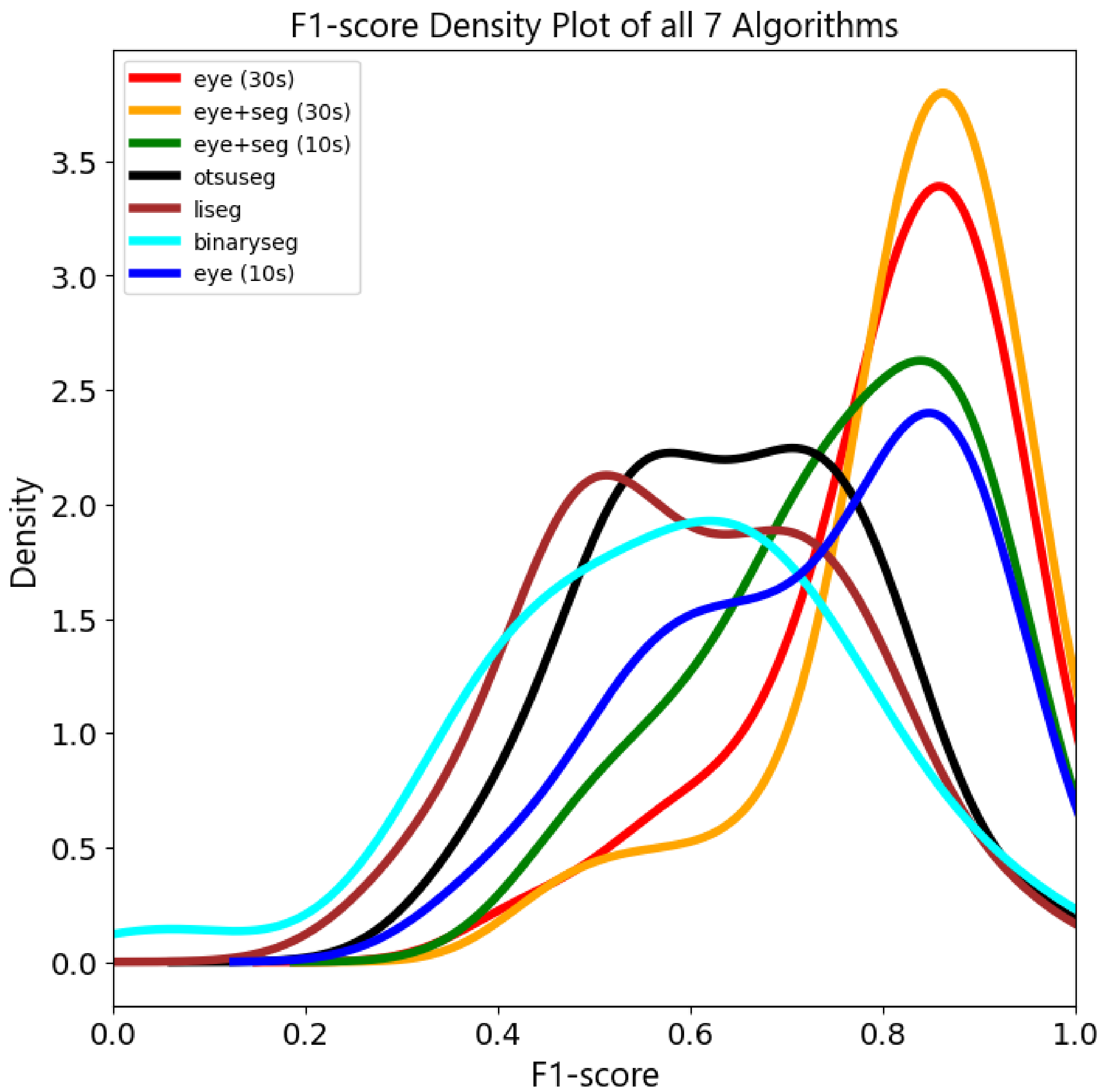


Figure 18: Density plot of the f1-score of all 30 images

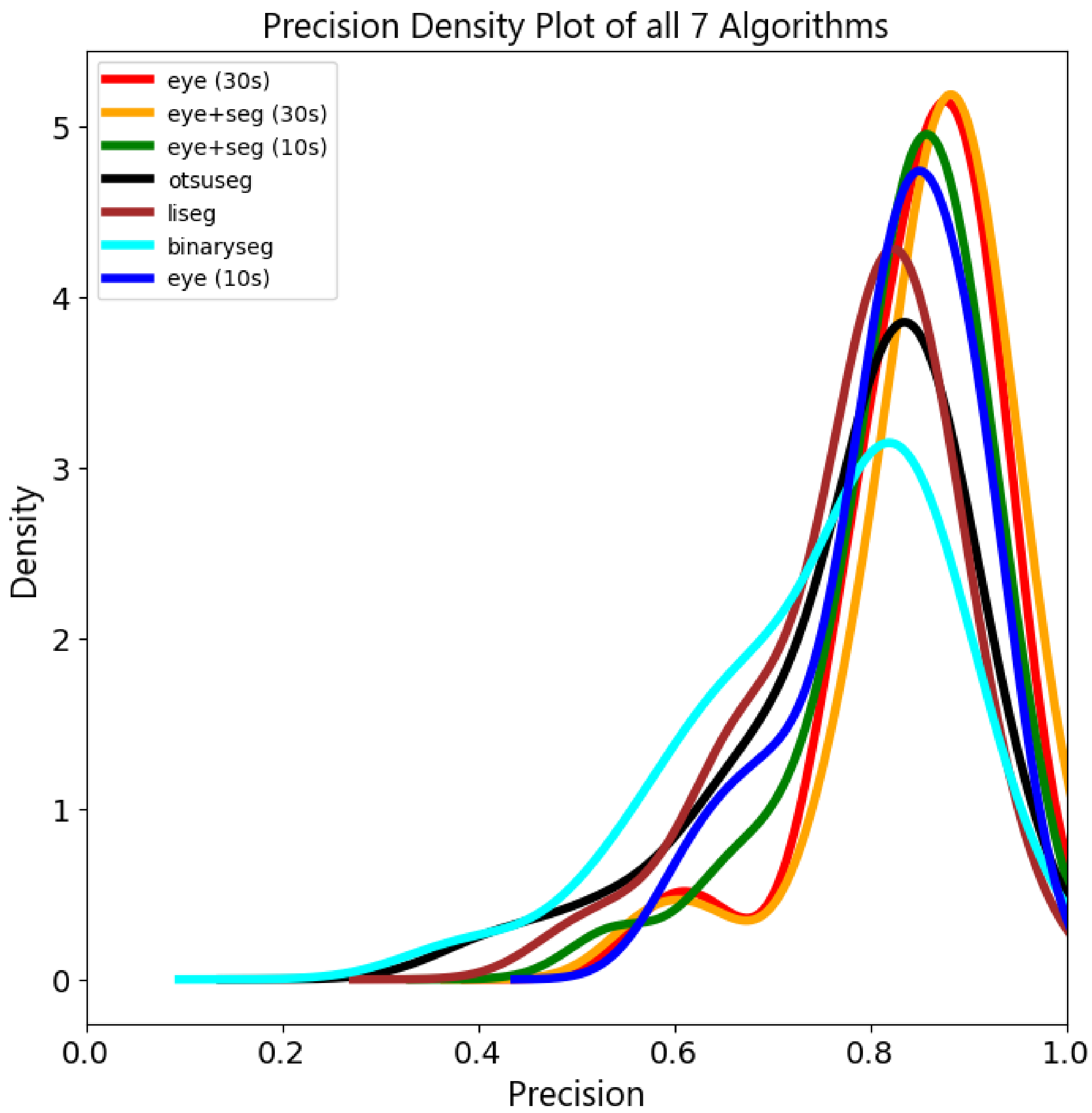


Figure 19: Density plot of the precision of all 30 images

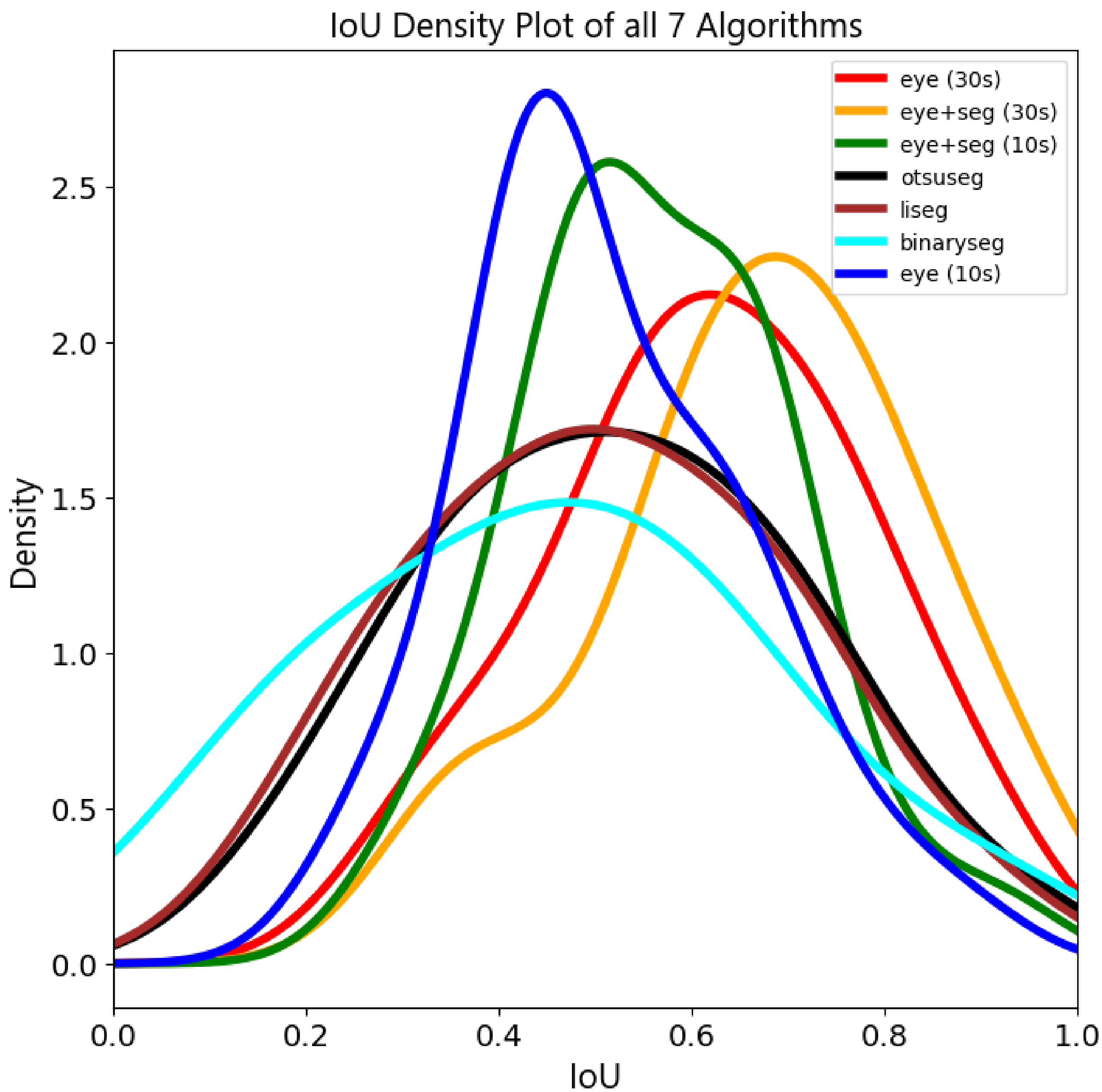


Figure 20: Density plot of the IoU of all 30 images

The density distribution and horizontal position of each metric curve (from left to right on the x-axis) characterize the performance of each algorithm with respect to the metric in question, according to the visual interpretation of all plots. The density distribution signifies the population of image samples that fall within a specific range of values for a metric. Model performance increases as its measure gets closer to 1.0, and the curve moves to the right on the horizontal axis. The density distribution increases as the curve amplitude increase as it distinguishes the density distribution. Figures 11 through 13 show that the accuracy, precision, and dice coefficient for the HITL 30 seconds eye tracking plus segmentation algorithm denoted as eye + seg (30s) generate a curve that is the furthest to the right on the horizontal axis and the densest population distribution. For all metrics, the HITL 10s seconds eye-tracking with segmentation comes in second, followed by the 30 seconds eye-tracking masking algorithm and the 10 seconds eye-tracking masking algorithm. Otsu, Li, and Binary thresholding—the other three traditional segmentation algorithms—performed quite poorly because they simultaneously have low-density distribution, and their curves are least displaced to the right.

Contrarily, the accuracy, precision, and f1-score do not entirely follow the same pattern as the density plot of the IoU. The standard mean and variance of each plot discussed below further support the findings from the plots.

Standard Mean and Variance of Density Plots

The density plots above are used to calculate the mean and variance for each metric and algorithm. The mean is described as the metric's central tendency across all 30 images, as determined by the distribution depicted by the density plots (M. S. Khan et al., 2006).

The variance of a metric from a density plot, on the other hand, provides a statistical assessment of how dispersed or spread out the values of that metric is from its mean or central tendency within a dataset. A density plot's variance reflects the metric's variability or uncertainty throughout the dataset (Kwon et al., 2020). The variance quantifies the average squared deviation of each result from the mean. A higher variance indicates a broader range of values and vice versa (Thrun et al., 2020). For this work, a model's performance is demonstrated by a reduced variance and a higher mean. The summary of each metric's standard mean and variance is tabulated below from Table 2-5.

Summary of Standard Mean and Variance of Algorithms for all 4 Metrics

Table 2: Mean and variance of algorithms of the f1-Score for all images.

Algorithms	Mean	Variance
Binary Thresholding	0.577777	0.035392
Li Thresholding	0.601782	0.024413
Otsu Segmentation	0.641472	0.020001
Eye tracking Masking (10s)	0.727934	0.024796
Eye tracking Masking (30s)	0.799737	0.017095
HITL Eye tracking + K-means Segmentation (10s)	0.755833	0.019198
HITL Eye tracking + K-means Segmentation (30s)	0.820695	0.015446

Table 3: Mean and variance of algorithms of the accuracy/weighted recall for all images.

Algorithms	Mean	Variance
Binary Thresholding	0.554981	0.031331
Li Thresholding	0.582996	0.022763
Otsu Segmentation	0.607348	0.021382
Eye tracking Masking (10s)	0.719600	0.028753
Eye tracking Masking (30s)	0.797470	0.018913
HITL Eye tracking + K-means Segmentation (10s)	0.742291	0.021863
HITL Eye tracking + K-means Segmentation (30s)	0.812758	0.016222

Table 4: Mean and variance of algorithms of the precision for all images.

Algorithms	**Mean**	**Variance**
Binary Thresholding	0.750424	0.017010
Li Thresholding	0.782473	0.010378
Otsu Segmentation	0.779873	0.015117
Eye tracking Masking (10s)	0.819132	0.007655
Eye tracking Masking (30s)	0.846352	0.007108
HITL Eye tracking + K-means Segmentation (10s)	0.829000	0.008378
HITL Eye tracking + K-means Segmentation (30s)	0.858022	0.007943

Table 5: Mean and variance of algorithms of the IoU for all images.

Algorithms	**Mean**	**Variance**
Binary Thresholding	0.456237	0.054320
Li Thresholding	0.508503	0.037192
Otsu Segmentation	0.520656	0.037552
Eye tracking Masking (10s)	0.512745	0.021052
Eye tracking Masking (30s)	0.617565	0.026748
HITL Eye tracking + K-means Segmentation (10s)	0.566304	0.019100
HITL Eye tracking + K-means Segmentation (30s)	0.668835	0.027230

Computational Time

The computational time of a segmentation algorithm is also critical when considering the efficiency of the robot de-caking system. After ten iterations using Python programming, the range, mean, and standard deviation of the running times for all the segmentation techniques for all 30 images were calculated and shown in Table 6.

Table 6: Average computational time of each segmentation algorithm for the test images.

Algorithms	**Running Time (seconds)**			
	Lower Limit	**Upper Limit**	**Mean**	**Standard Deviation**
Binary Thresholding	0.555665	0.626781	0.579662	0.020571
Li Thresholding	0.559737	0.610923	0.581879	0.017935
Otsu Segmentation	0.557458	0.583873	0.571006	0.008670
Eye tracking (10s)	0.537062	0.569525	0.554724	0.008329
Eye tracking (30s)	0.527560	0.548425	0.540755	0.006566
HITL Eye tracking + K-means Segmentation (10s)	0.524034	0.533959	0.528978	0.003315
HITL Eye tracking + K-means Segmentation (30s)	0.497285	0.5258369	0.510655	0.008387

CHAPTER V

CONCLUSION AND FUTURE WORK

This study proposes a HITL eye-tracking plus segmentation algorithm to assist the automated robot de-caking system for powder-bed AM. The experiment's findings demonstrate how much better the semi-supervised HITL eye-tracking with segmentation model performs than the traditional segmentation methods currently in use. Additionally, this framework provides the opportunity to incorporate biological data about humans to increase the adaptability and effectiveness of artificial intelligence algorithms and develop the ideas of human-autonomy teaming and convergent production in advanced manufacturing. With an average accuracy, f1-score, precision, and IoU score of 81.2%, 82.3%, 85.8%, and 66.9%, respectively, the suggested HITL 30 seconds eye tracking plus segmentation has the best performance out of all the algorithms evaluated and compared.

Similarly, the proposed HITL framework (for both 30 and 10 seconds eye tracking with segmentation) falls within the same range of values of the computational or mean running times tested classic segmentation algorithms, even though it performs better.

New directions of research can be developed based on the proposed framework. HITL framework lays the foundation for further research as other machine learning algorithms, such as Bayesian and deep learning, could be applied. The concept of human-in-the-loop is still in the early stages, and this work promotes further the idea of human autonomy teaming, which contributes to the futuristic industry 5.0.

Variations of this framework could be enhanced to support video-type data and not only image-type datasets, making it more useful for a broader range of applications. Also, the time spent watching the eye movement must coincide with the time spent processing the video data in practical applications, even though the segmentation algorithm's time requirements for the automated de-caking procedure are practical.

REFERENCES

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. E., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938. https://doi.org/10.1016/J.HELIYON.2018.E00938

Akbay, Ö. C., Bahçe, E., Uysal, A., & Gezer, İ. (2022). Productıon and Cleaning of Lattice Structures Used in the Space and Aerospace Industry with Metal Additive Manufacturing Method. *Journal of Materials Engineering and Performance*, *31*(8), 6310–6321. https://doi.org/10.1007/S11665-021-06541-2/FIGURES/19

Arrizubieta, J. I., Ukar, O., Ostolaza, M., & Mugica, A. (2020). Study of the Environmental Implications of Using Metal Powder in Additive Manufacturing and Its Handling. *Metals 2020, Vol. 10, Page 261*, *10*(2), 261. https://doi.org/10.3390/MET10020261

Asare-Manu, V., Karunathilake, S., & Geng, Z. (2023). Image Segmentation with Human-in-the-loop In Automated De-caking Process for Powder Bed Additive Manufacturing. *ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Bali, A., & Singh, S. N. (2015). A review on the strategies and techniques of image segmentation. *International Conference on Advanced Computing and Communication Technologies, ACCT*, *2015-April*, 113–120. https://doi.org/10.1109/ACCT.2015.63

Bertol, L. S., Júnior, W. K., Silva, F. P. da, & Aumund-Kopp, C. (2010). Medical design: Direct metal laser sintering of Ti–6Al–4V. *Materials & Design*, *31*(8), 3982–3988. https://doi.org/10.1016/J.MATDES.2010.02.050

Bovik, A. C. (2009). Basic Binary Image Processing. *The Essential Guide to Image Processing*, 69–96. https://doi.org/10.1016/B978-0-12-374457-9.00004-4

Cai, C., Tey, W. S., Chen, J., Zhu, W., Liu, X., Liu, T., Zhao, L., & Zhou, K. (2021). Comparative study on 3D printing of polyamide 12 by selective laser sintering and multi jet fusion. *Journal of Materials Processing Technology, 288*, 116882. https://doi.org/10.1016/J.JMATPROTEC.2020.116882

Cao, Q., Qingge, L., & Yang, P. (2021). Performance Analysis of Otsu-Based Thresholding Algorithms: A Comparative Study. *Journal of Sensors*, *2021*. https://doi.org/10.1155/2021/4896853

Chai, C., & Li, G. (2020). *Human-in-the-loop Techniques in Machine Learning*.

Chakraborty, S., Roy, M., & Hore, S. (2017). *A Study on Different Edge Detection Techniques in Digital Image Processing*. 100–122. https://doi.org/10.4018/978-1-5225-1025-3.CH005:

Chhabra, G., Vashisht, V., & Ranjan, J. (2017). A Comparison of Multiple Imputation Methods for Data with Missing Values. *Indian Journal of Science and Technology*. https://doi.org/10.17485/ijst/2017/v10i19/110646

Cuellar, J. S., Smit, G., Plettenburg, D., & Zadpoor, A. (2018). Additive manufacturing of non-assembly mechanisms. *Additive Manufacturing*, *21*, 150–158. https://doi.org/10.1016/J.ADDMA.2018.02.004

Daniel, P., Raju, R., & Neelima, G. (2012). *Image Segmentation by using Histogram Thresholding*.

Dutta, K., Talukdar, D., & Bora, S. S. (2022). Segmentation of unhealthy leaves in cruciferous crops for early disease detection using vegetative indices and Otsu thresholding of aerial images. *Measurement*, *189*, 110478. https://doi.org/10.1016/J.MEASUREMENT.2021.110478

G, N. V, & S, H. K. (2011). Study and comparison of various image edge detection techniques used in quality inspection and evaluation of agricultural and food products by computer vision. *International Journal of Agricultural and Biological Engineering*, *4*(2), 83–90. https://doi.org/10.25165/IJABE.V4I2.277

Geng, Z., & Bidanda, B. (2020). Medical Applications of Additive Manufacturing. *Bio-Materials and Prototyping Applications in Medicine: Second Edition*, 97–110. https://doi.org/10.1007/978-3-030-35876-1_6/FIGURES/3

Geng, Z., & Bidanda, B. (2022). Additively Manufactured Dentures, Crowns, and Bridges. *Additive Manufacturing in Biomedical Applications*, 472–478. https://doi.org/10.31399/ASM.HB.V23A.A0006899

Hsiao, Y. T., Chuang, C. L., Jiang, J. A., & Chien, C. C. (2005). A contour based image segmentation algorithm using morphological edge detection. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *3*, 2962–2967. https://doi.org/10.1109/ICSMC.2005.1571600

Huang, J., Qin, Q., & Wang, J. (2020). A Review of Stereolithography: Processes and Systems. *Processes 2020, Vol. 8, Page 1138*, *8*(9), 1138. https://doi.org/10.3390/PR8091138

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(03), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jiang, F., Frater, M. R., & Pickering, M. (2012). Threshold-based image segmentation through an improved particle swarm optimisation. *2012 International Conference on Digital Image Computing Techniques and Applications, DICTA 2012*. https://doi.org/10.1109/DICTA.2012.6411743

K, S., & S, S. (2014). Review on Classification Based on Artificial Neural Networks. *The International Journal of Ambient Systems and Applications*, *2*(4), 11–18. https://doi.org/10.5121/IJASA.2014.2402

Kang, C. C., & Wang, W. J. (2007). A novel edge detection method based on the maximizing objective function. *Pattern Recognition*, *40*(2), 609–618. https://doi.org/10.1016/J.PATCOG.2006.03.016

Karoui, I., Fablet, R., Boucher, J. M., & Augustin, J. M. (2007). Unsupervised region-based image segmentation using texture statistics and level-set methods. *2007 IEEE International Symposium on Intelligent Signal Processing, WISP*. https://doi.org/10.1109/WISP.2007.4447617

Kazemi, F. M., Akbarzadeh-T, M. R., Rahati, S., & Rajabi, H. (2008). Fast image segmentation using C-means based Fuzzy Hopfield neural network. *Canadian Conference on Electrical and Computer Engineering*, 1855–1859. https://doi.org/10.1109/CCECE.2008.4564866

Khan, M. S., Coulibaly, P., & Dibike, Y. (2006). Uncertainty analysis of statistical downscaling methods. *Journal of Hydrology*, *319*(1–4), 357–382. https://doi.org/10.1016/J.JHYDROL.2005.06.035

Khan, W. (2014). *A Survey: Image Segmentation Techniques Article in International Journal of Future Computer and Communication*. https://doi.org/10.7763/IJFCC.2014.V3.274

Kruth, J. P., Mercelis, P., Van Vaerenbergh, J., Froyen, L., & Rombouts, M. (2005). Binding mechanisms in selective laser sintering and selective laser melting. *Rapid Prototyping Journal*, *11*(1), 26–36. https://doi.org/10.1108/13552540510573365/FULL/PDF

Kruth, J. P., Wang, X., Laoui, T., & Froyen, L. (2003). Lasers and materials in selective laser sintering. *Assembly Automation*, *23*(4), 357–371. https://doi.org/10.1108/01445150310698652/FULL/PDF

Kumbhar, N. N., & Mulay, A. V. (2018). Post Processing Methods used to Improve Surface Finish of Products which are Manufactured by Additive Manufacturing Technologies: A Review. *Journal of The Institution of Engineers (India): Series C*, *99*(4), 481–487. https://doi.org/10.1007/S40032-016-0340-Z/FIGURES/3

Kurmi, Y., & Chaurasia, V. (2018). Multifeature-based medical image segmentation. *IET Image Processing*, *12*(8), 1491–1498. https://doi.org/10.1049/IET-IPR.2017.1020

Kwon, Y., Won, J. H., Kim, B. J., & Paik, M. C. (2020). Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, *142*, 106816. https://doi.org/10.1016/J.CSDA.2019.106816

Lakshmi, S., & Sankaranarayanan, V. (2010). Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications. *IJCA Special Issue On*.

Li, C. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, *26*(4), 617–625. https://doi.org/10.1016/0031-3203(93)90115-D

Mary Synthuja Jain Preetha, M., Padma Suresh, L., & John Bosco, M. (2012). Image segmentation using seeded region growing. *2012 International Conference on Computing, Electronics and Electrical Technologies, ICCEET 2012*, 576–583. https://doi.org/10.1109/ICCEET.2012.6203897

Mohamed, O. A., Masood, S. H., & Bhowmik, J. L. (2015). Optimization of fused deposition modeling process parameters: a review of current research and future prospects. *Advances in Manufacturing*, *3*(1), 42–53. https://doi.org/10.1007/S40436-014-0097-7/METRICS

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., Fernández-Leal, Á., & Mosqueira-Rey, E. (123 C.E.). *Human-in-the-loop machine learning: a state of the art-in-the-loop machine learning · Active learning · Interactive machine learning · Machine teaching · Curriculum learning · Explainable AI*. https://doi.org/10.1007/s10462-022-10246-w

Nguyen, H., Adrian, N., Lim, J., Yan, X., Salfity, J. M., Allen, W., & Pham, Q.-C. (2020). Development of a Robotic System for Automated Decaking of 3D-Printed Parts; Development of a Robotic System for Automated Decaking of 3D-Printed Parts. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.0/Linux-x86_64

Ohsawa, M. (2009). [Heavy metal-induced immunotoxicity and its mechanisms]. *Yakugaku Zasshi : Journal of the Pharmaceutical Society of Japan*, *129*(3), 305–319. https://doi.org/10.1248/YAKUSHI.129.305

Rahman, M. A., & Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10072 LNCS*, 234–244. https://doi.org/10.1007/978-3-319-50835-1_22/FIGURES/3

Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., Hamid, T., Islam, M. T., Kashem, S., Mahbub, Z. Bin, Ayari, M. A., & Chowdhury, M. E. H. (2020). Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *IEEE Access*, *8*, 191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384

Raut, S., Raghuvanshi, M., Dharaskar, R., & Raut, A. (2009). Image Segmentation – A State-Of-Art Survey for Prediction; Image Segmentation – A State-Of-Art Survey for Prediction. *2009 International Conference on Advanced Computer Control*. https://doi.org/10.1109/ICACC.2009.78

Roth, G. A., Geraci, C. L., Stefaniak, A., Murashov, V., & Howard, J. (2019). Potential occupational hazards of additive manufacturing. *Https://Doi.Org/10.1080/15459624.2019.1591627*, *16*(5), 321–328. https://doi.org/10.1080/15459624.2019.1591627

Russo, F., & Lazzari, A. (2005). Color edge detection in presence of Gaussian noise using nonlinear prefiltering. *IEEE Transactions on Instrumentation and Measurement*, *54*(1), 352–358. https://doi.org/10.1109/TIM.2004.834074

Sahoo, P. K., Soltani, S., & Wong, A. K. C. (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, *41*(2), 233–260. https://doi.org/10.1016/0734-189X(88)90022-9

Salihah Abdul-nasir, A., Yusoff Mashor, M., & Mohamed, Z. (2013). *Colour Image Segmentation Approach for Detection of Malaria Parasites Using Various Colour Models and k-Means Clustering*.

Sevak, J. S., Kapadia, A. D., Chavda, J. B., Shah, A., & Rahevar, M. (2018). Survey on semantic image segmentation techniques. *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, 306–313. https://doi.org/10.1109/ISS1.2017.8389420

Shukla, S. (2014). A Review ON K-means DATA Clustering APPROACH. *International Journal of Information & Computation Technology*, *4*, 1847–1860. http://www.irphouse.com

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, *15*(1), 1–28. https://doi.org/10.1186/S12880-015-0068-X/TABLES/5

Tahmina, T., Garcia, M., Geng, Z., & Bidanda, B. (2023). A Survey of Smart Manufacturing for High-Mix Low-Volume Production in Defense and Aerospace Industries. *Lecture Notes in Mechanical Engineering*, 237–245. https://doi.org/10.1007/978-3-031-18326-3_24/FIGURES/1

Terry, P. J., & Vu, D. (1993). Edge detection using neural networks. *Conference Record of the Asilomar Conference on Signals, Systems & Computers*, *1*, 391–395. https://doi.org/10.1109/ACSSC.1993.342541

Thrun, M. C., Gehlert, T., & Ultsch, A. (2020). Analyzing the fine structure of distributions. *PLOS ONE*, *15*(10), e0238835. https://doi.org/10.1371/JOURNAL.PONE.0238835

Van Der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in python. *PeerJ*, *2014*(1), e453. https://doi.org/10.7717/PEERJ.453/FIG-5

Van Der Zwaag, J., Slump, K., & Spaanenburg, L. (n.d.). *Analysis of Neural Networks for Edge Detection*.

Vayre, B., Vignat, F., & Villeneuve, F. (2012). Metallic additive manufacturing: state-of-the-art review and prospects. *Mechanics & Industry*, *13*(2), 89–96. https://doi.org/10.1051/MECA/2012003

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, *2018*. https://doi.org/10.1155/2018/7068349

Wahab, A., Sharif, M., Shah, J. H., Mohsin, S., & Raza, M. (2013). Sub-Holistic Hidden Markov Model for Face Recognition mudassar raza Related papers A Survey on Face Det ect ion and Recognit ion Approaches Sub-Holistic Hidden Markov Model for Face Recognition. *Research Journal of Recent Sciences*, *2*(5), 10–14. www.isca.in

Wang, Z., Wang, E., & Zhu, Y. (2020). Image segmentation evaluation: a survey of methods. *Artificial Intelligence Review*, *53*(8), 5637–5674. https://doi.org/10.1007/S10462-020-09830-9/FIGURES/5

Wesolkowski, S., & Fieguth, P. (2002). A Markov random fields model for hybrid edge- and region-based color image segmentation. *Canadian Conference on Electrical and Computer Engineering*, *2*, 945–949. https://doi.org/10.1109/CCECE.2002.1013070

Wu, K., & Ban, T. (2011). Optimal threshold image segmentation method based on genetic algorithm in wheel set online measurement. *Proceedings - 3rd International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2011*, *2*, 799–802. https://doi.org/10.1109/ICMTMA.2011.483

Wu, Q., & Castleman, K. R. (2023). Image Segmentation. *Microscope Image Processing, Second Edition*, 119–152. https://doi.org/10.1016/B978-0-12-821049-9.00003-4

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, *135*, 364–381. https://doi.org/10.1016/J.FUTURE.2022.05.014

Xie, G., & Lu, W. (2013). Image Edge Detection Based On Opencv. *International Journal of Electronics and Electrical Engineering*. https://doi.org/10.12720/ijeee.1.2.104-106

Xu, A., Wang, L., Feng, S., & Qu, Y. (2010). Threshold-based level set method of image segmentation. *Proceedings - 3rd International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2010*, 703–706. https://doi.org/10.1109/ICINIS.2010.181

Xu, X., Xu, S., Jin, L., & Song, E. (2011). Characteristic analysis of Otsu threshold and its applications. *Pattern Recognition Letters*, *32*(7), 956–961. https://doi.org/10.1016/J.PATREC.2011.01.021

Yap, C. Y., Chua, C. K., Dong, Z. L., Liu, Z. H., Zhang, D. Q., Loh, L. E., & Sing, S. L. (2015). Review of selective laser melting: Materials and applications. *Applied Physics Reviews*, *2*(4), 041101. https://doi.org/10.1063/1.4935926

Yu, X., & Yla-Jaaski, J. (1991). A new algorithm for image segmentation based on region growing and edge detection. *Proceedings - IEEE International Symposium on Circuits and Systems*, *1*, 516–519. https://doi.org/10.1109/ISCAS.1991.176386

Zaim, A. (2008). An edge-based approach for segmentation of prostate ultrasouind images using phase symmetry. *2008 3rd International Symposium on Communications, Control, and Signal Processing, ISCCSP 2008*, 10–13. https://doi.org/10.1109/ISCCSP.2008.4537183

Zhang, H., Fritts, J. E., & Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, *110*(2), 260–280. https://doi.org/10.1016/J.CVIU.2007.08.003

Zhang, R., Zhao, G., & Su, L. (2005). A new edge detection method in image processing. *ISCIT 2005 - International Symposium on Communications and Information Technologies 2005, Proceedings*, *II*, 430–433. https://doi.org/10.1109/ISCIT.2005.1566889

Zhang, X., & Tay, A. L. P. (2007). Fast Learning Artificial Neural Network (FLANN) based color image segmentation in R-G-B-S-V cluster space. *IEEE International Conference on Neural Networks - Conference Proceedings*, 563–568. https://doi.org/10.1109/IJCNN.2007.4371018

Zhou, Y. M., Jiang, S. Y., & Yin, M. L. (2008). A region-based image segmentation method with mean-shift clustering algorithm. *Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008*, *2*, 366–370. https://doi.org/10.1109/FSKD.2008.363

Ziaee, M., & Crane, N. B. (2019). Binder jetting: A review of process, materials, and methods. *Additive Manufacturing*, *28*, 781–801. https://doi.org/10.1016/J.ADDMA.2019.05.031

APPENDIX

APPENDIX

This section contains the complete set of all 30 images analyzed by all the classic and existing algorithms. From these images, it can be noticed that none of these classic segmentation algorithms works well (separating the foreground from the background and creating a distinct powder/non-powder segment) for all or most of the images, setting the motivation and basis for our research.

Figure 21: All 30 segmented images for Otsu thresholding

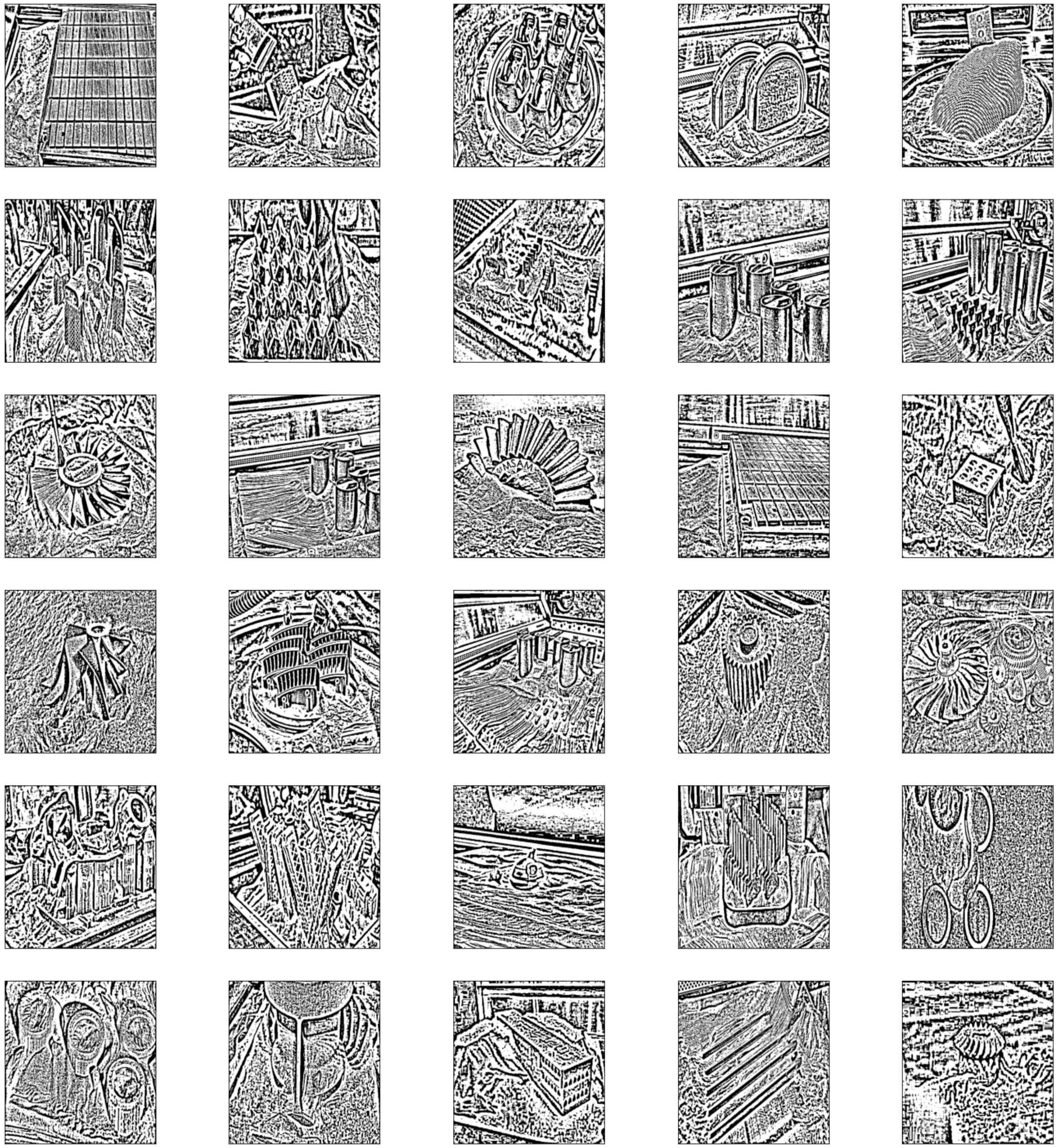


Figure 22: All 30 segmented images for Niblack thresholding

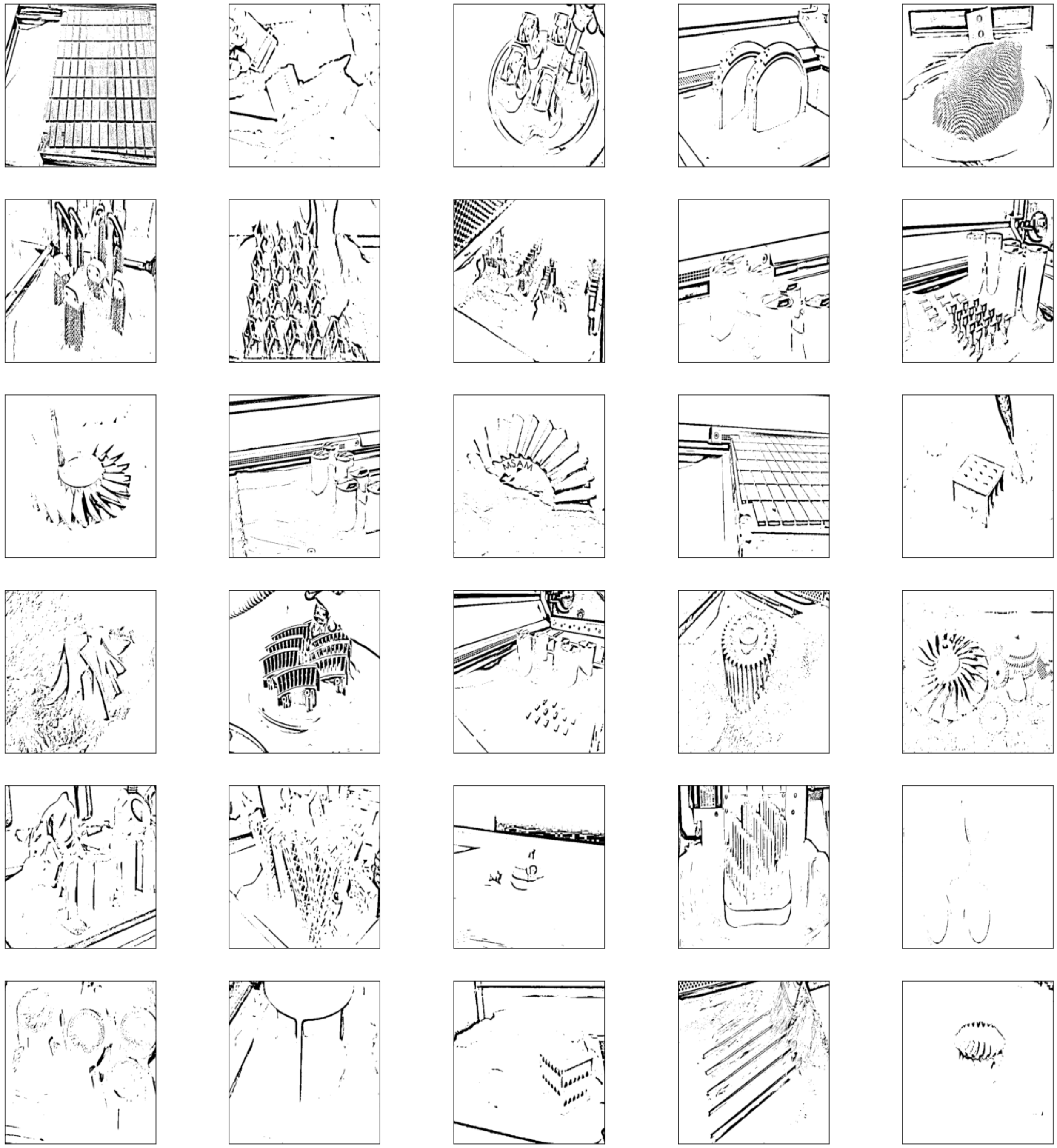


Figure 23: All 30 segmented images for Sauvola thresholding

Figure 24: All 30 segmented images for Otsu thresholding

Figure 25: All 30 segmented images for Li iterative thresholding

Figure 26: All 30 segmented images for to-zero thresholding

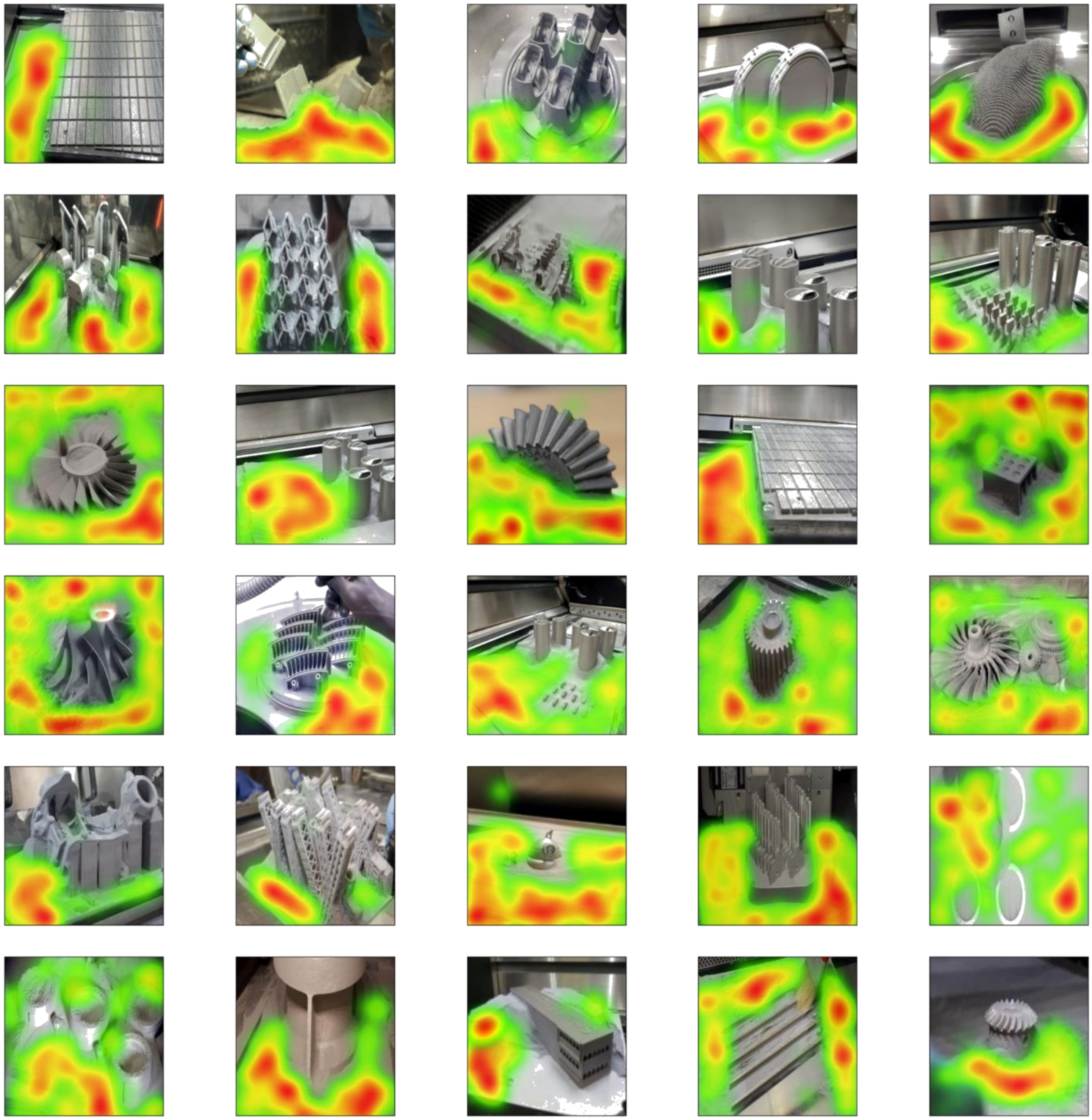


Figure 27: All heatmap images from 30 seconds of eye tracking

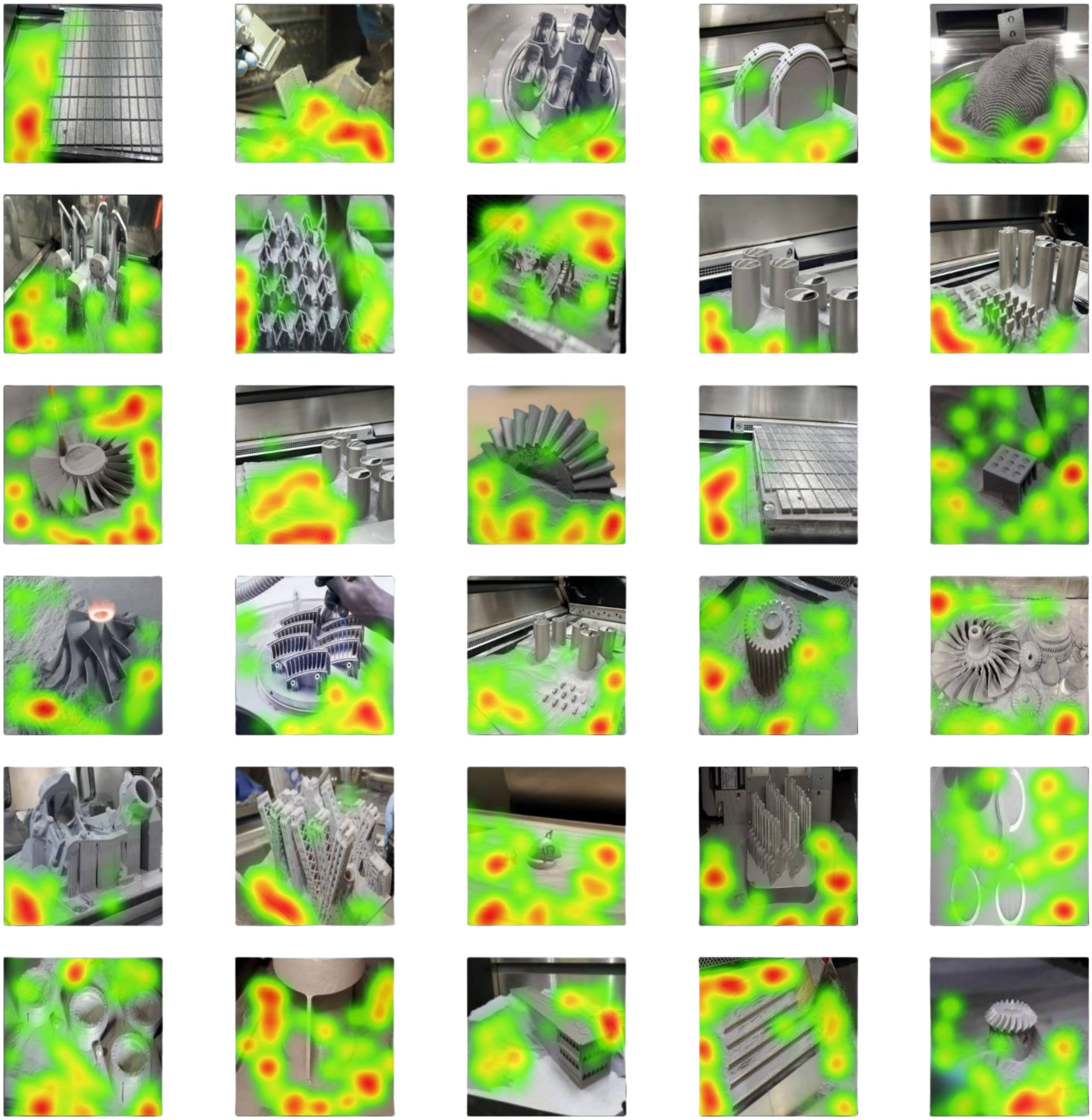


Figure 28: All heatmap images from 10 seconds of eye tracking

Figure 29: All color images from the proposed HITL K-means plus 30 seconds of eye tracking

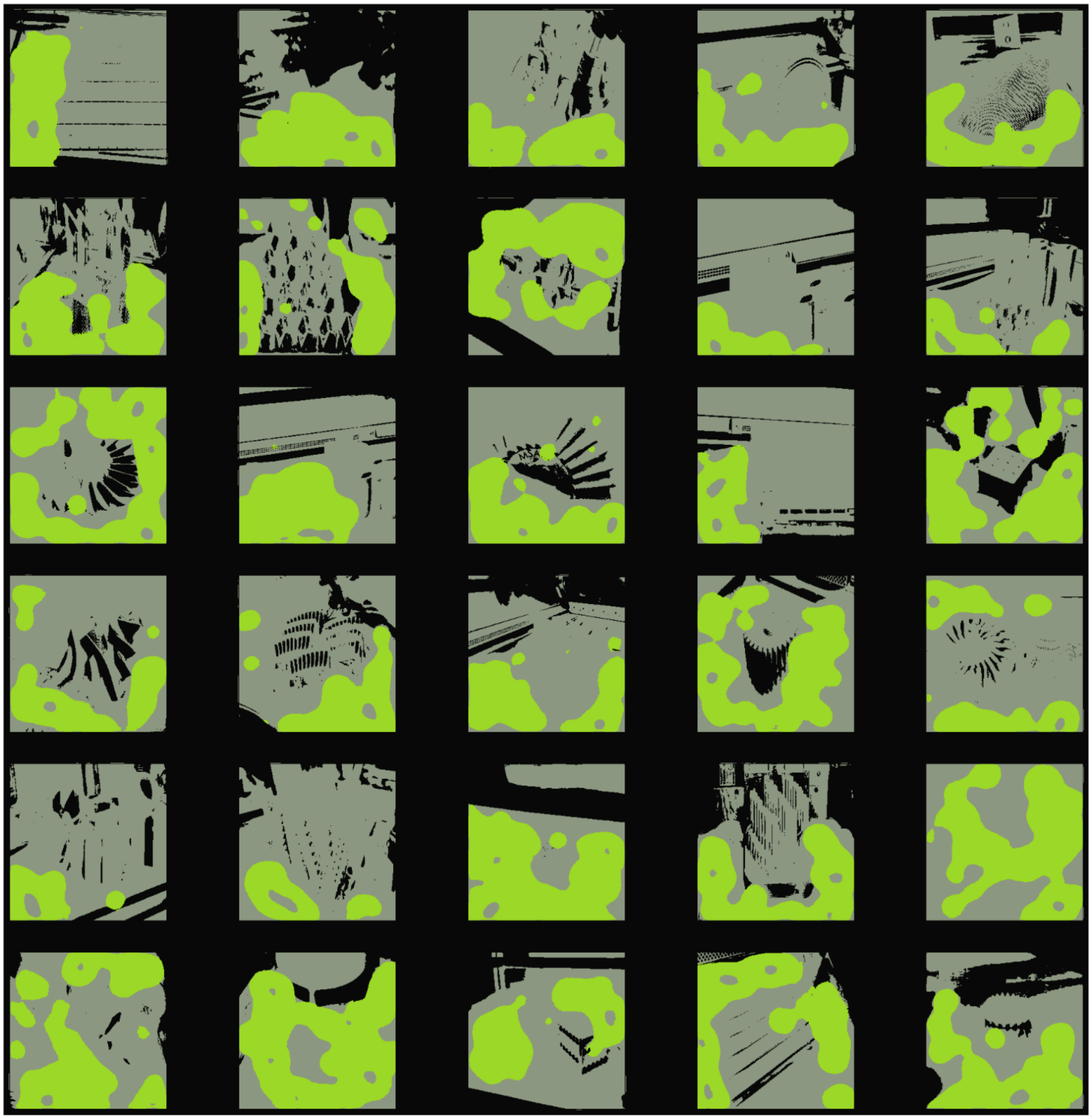


Figure 30: All color images from K-means plus 10 seconds of eye tracking

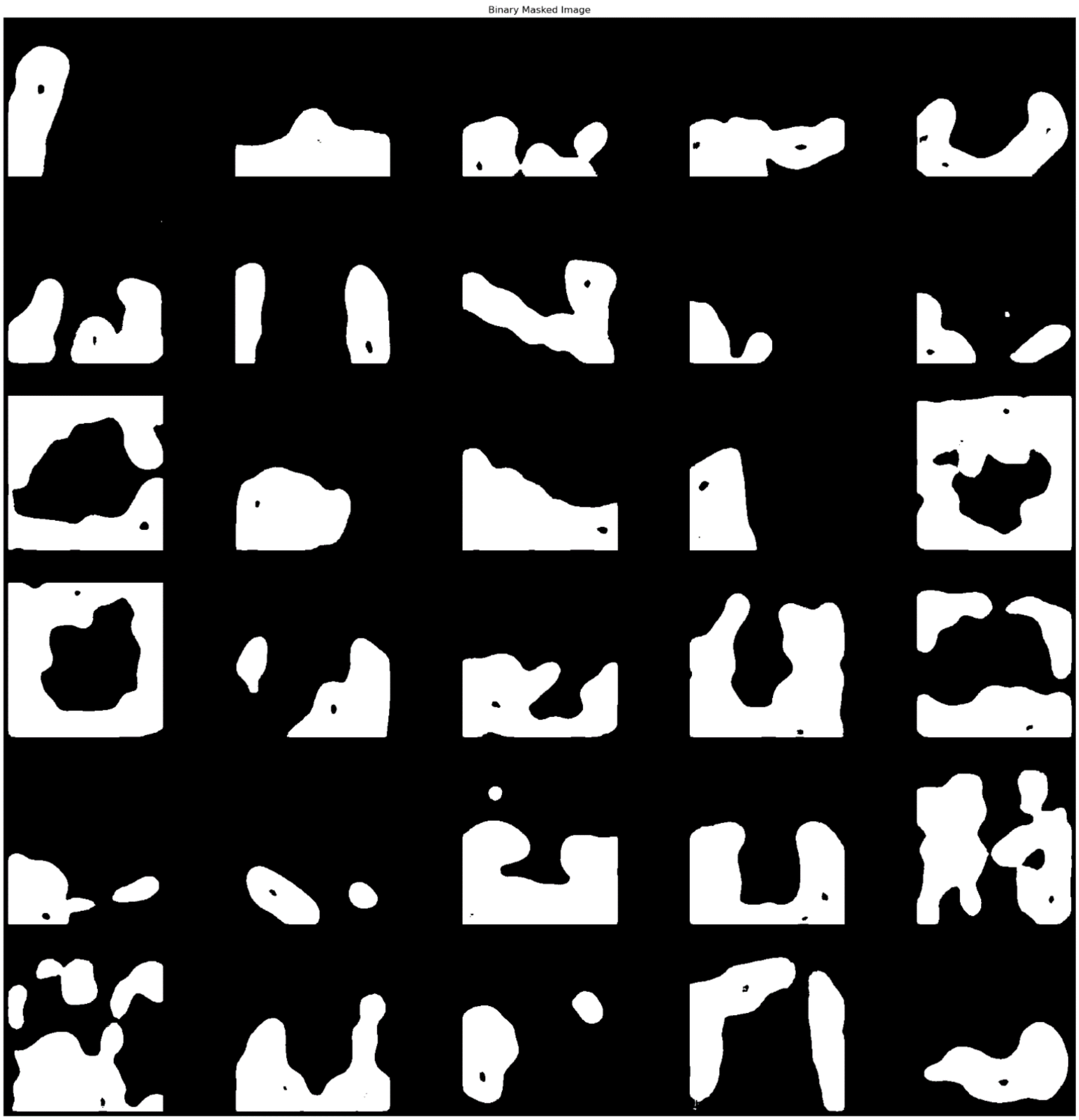


Figure 31: All black and white (binary) images from the proposed HITL K-means plus 30 seconds of eye tracking segmentation algorithm

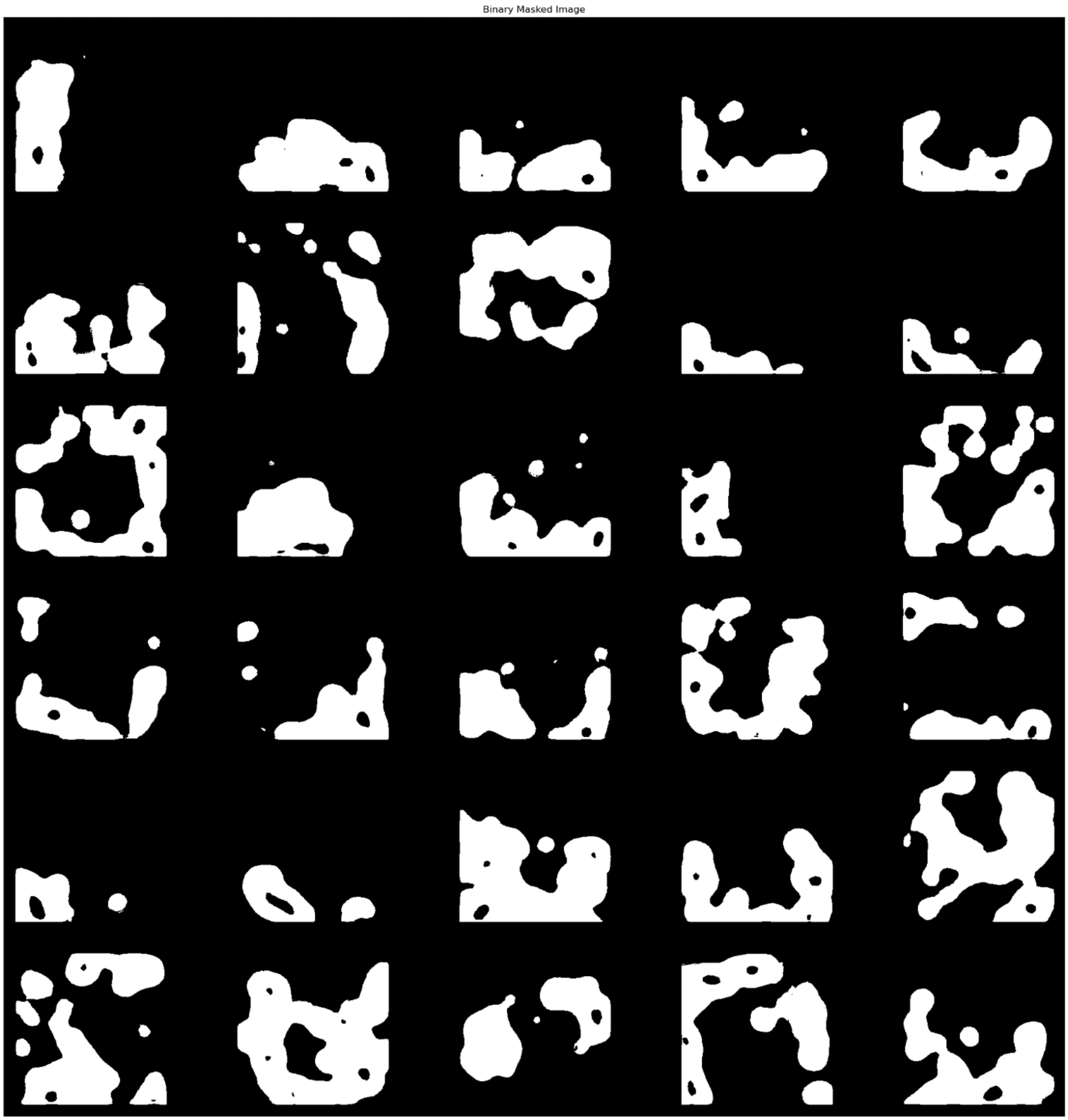


Figure 32: All black and white (binary) images from the HITL K-means plus 10 seconds of eye tracking segmentation algorithm

BIOGRAPHICAL SKETCH

Vincent Opare Addo Asare-Manu acquired his bachelor's degree in chemical engineering from the Kwame Nkrumah University of Science and Technology in Ghana in 2019. He joined the Manufacturing and Industrial Engineering department at The University of Texas Rio Grande Valley and graduated with a Master of Science in Manufacturing Engineering in July 2023. Through the UTRGV Office of the Vice President of Research via The Office of Technology Commercialization (OTC), Vincent was awarded a full two-year scholarship as a Graduate Assistantship, which sponsored his entire education at UTRGV. Through his exceptional academic performance and good standing with a GPA of 4.0/4.0, Vincent earned some other institutional awards, such as the Archibald & Martaret McColl Scholarship and the Dean's Scholarship Award for multiple semesters. Also, with the support of the US Department of Defense (DoD) I-Dream4D Consortium, Vincent worked as a Research Assistant under the supervision of Dr. Zhaohui Geng. Vincent can be reached via email at vincent.asaremanu@gmail.com.