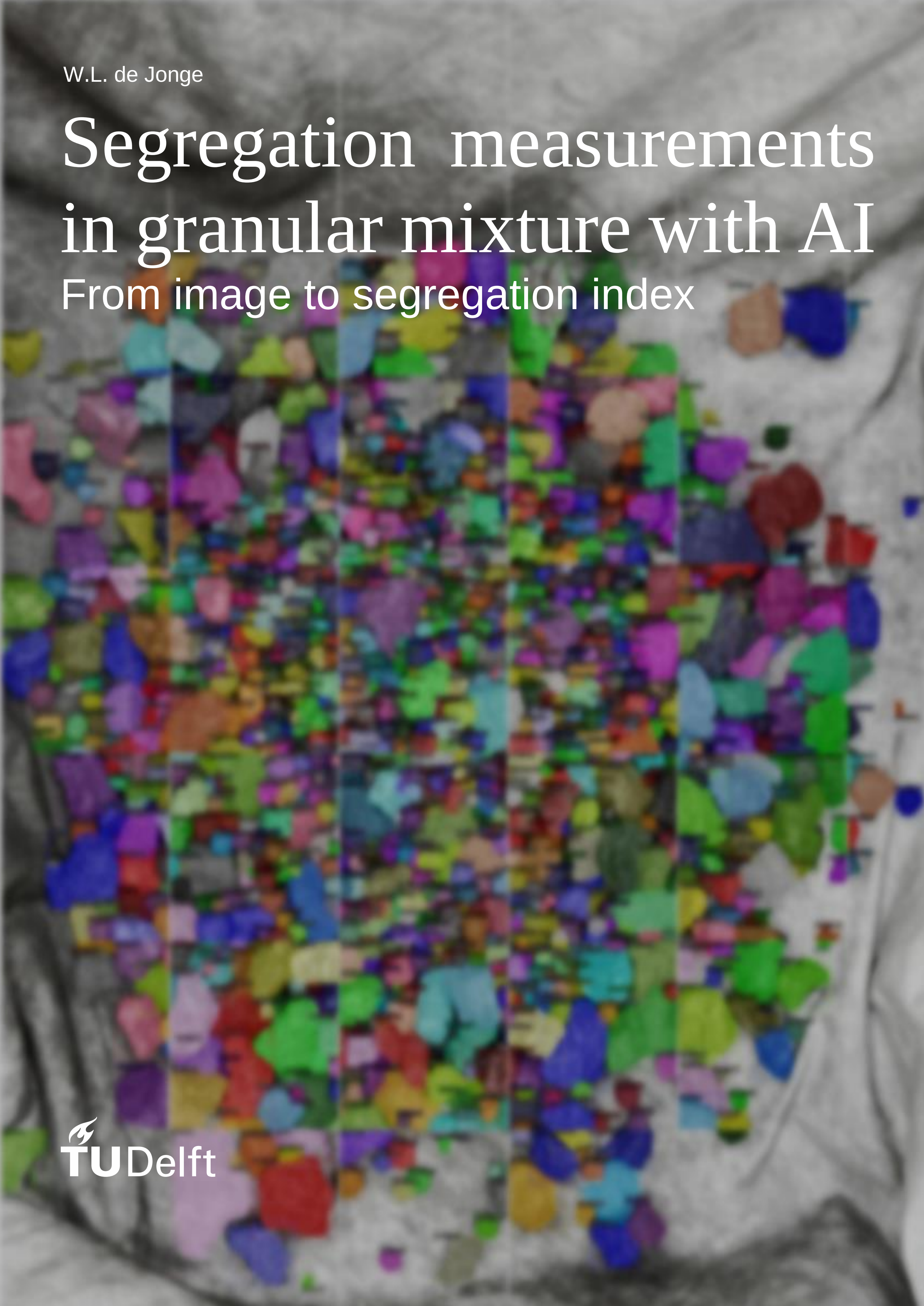
W.L. de Jonge

Segregation measurements in granular mixture with AI

From image to segregation index



Segregation measurements in granular mixture with AI

From image to segregation index

By

W.L. de Jonge

Master Thesis

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of
Delft University of Technology
to be defended publicly on Wednesday August 28, 2024 at 1:00 PM

Student number:	4466500	
MSc track:	Multi-Machine Engineering	
Report number:	2024.MME.8946	
Thesis committee:	Prof. dr. ir. D.L. Schott,	TU Delft committee Chair, ME
	Dr. ir. Y. Pang,	TU Delft committee member, ME
	A.H. Hadi,	TU Delft committee member, ME
Date:	August 5, 2024	

Preface

My interest in artificial intelligence was initially piqued by its increasing use across various industries and the ongoing conversations about its potential impact. The opportunity to delve into AI through this thesis has allowed me to connect my personal interests with academic research. Despite being relatively new to the realms of AI, machine learning, and deep learning, this project helps as an introductory journey into these fields, covering foundational principles and insights.

The focus of AI application in this thesis revolves around a hybrid approach aimed at simplifying measurement interpretation. The objective is to deepen our comprehension of particle segregation, a common yet often overlooked phenomenon in everyday scenarios. By blending elements of computer science and particle science, this project brings together two distinct disciplines to shed new light on this subject.

I would like to express my gratitude to prof. dr..ir. D. L. Schott for providing the opportunity to work on a project centred around AI. Additionally, I extend my thanks to dr. ir. Y. Pang for his supervision A. H. Hadi for his guidance with data collection and writing, as well as his patience throughout the process. Special thanks to G&G for their support and encouragement, helping me see this thesis through to completion.

I hope this thesis inspires further advancements in the use of AI for measuring particle mixture compositions and provides an example for the possibilities.

Summary

This report explores the possibilities for measuring segregation in granular material with the assistance of artificial intelligence (AI), a topic of interest for various industries handling particles including pharmaceuticals, agriculture and steel production. The ability to accurately measure the segregation of the granular materials can enhance quality control, efficiency, environmental impact, and safety. AI is introduced to allow more flexibility in the measuring of the segregation, which is crucial to creating a better understanding of the segregation process.

The reports commence with a literature review of the current methods of measuring and quantifying segregation in granular materials. Followed up by an introduction to AI and the ability to let a computer find the particles. Despite segregation being a decades old phenomenon, a method of extracting data without interacting with the mixture proves to be challenging. Requiring either to disturb the mixture with intrusive measuring methods or colouring particles and stable lighting for the non-intrusive methods. Both options limit the practicality of the measuring and therefore they are constrained within laboratories.

The primary objective of this report is to evaluate and compare techniques for measuring the segregation in granular material, which is split up into recognizing the particles and measuring the segregation based on the recognition. The focus lays on improving the accessibility for taking measurements in uncontrolled scenarios. This report looks at measuring the segregation of the material commonly found in a blast furnace. These materials are coke, sinter, and pellet. These three materials possess familiar characteristics in appearance and are troublesome to colour. Making them a challenge to show the capabilities of the AI's.

A comprehensive review and experimental comparison of several techniques was conducted, including sampling, image analysis, and machine learning algorithms. Data was collected from various samples of a three component, coke, sinter and pellet, mixture in the form of images. The performance of each technique was assessed based on their capabilities of recognizing the particles within images. The best performing technique was then applied to measure the segregation.

The findings show that AIs are capable of recognizing the coke, sinter, and pellet particles within the mixture. Since the AIs are recognizing the particles individually, the possibilities of measuring both material and size segregation open up. Resulting in the AI processing an image in such a manner that both segregations could be analysed. These findings show the potential of applying AI for recognizing the particles. The AI offers a comprehensive analysis of the mixture to improve the understanding of the composition and what is measured. However, the visualizations of the recognitions expose some weak points of the showcased technique, which allows for further improvements to the application.

In conclusion, the study highlights the potential of AI to improve the identification of granular materials to allow for segregation measurements. This has significant implications for various applications, where the materials or inconsistent lighting are challenging for conventional methods.

List of symbols

a	Neuron number a
a_m^L	Neuron in layer L with number m
A	Arithmetic mean
ab	Weighted biases between neuron a and neuron b
b	Neuron number b in the previous layer
β	Value between 0 and infinity to shift the F-score
C_i	Concentration of sample i
C_0	Concentration of sample for an ideal mixture
H	Harmonic mean
L	Layer number
n	Pixels in the width
n_i	Number of samples with concentration C_i
n	Number of samples
N	Number of samples in SI
N_p	Number of particles
m	Pixels in the height
M	Mixing index
σ	Standard deviation
σ^2	Variance
σ_0	Variance of a completely segregated system
σ_R	Variance of perfectly random system
p	Kernel size
p_c	Concentration of sample on location 1
P_c	Concentration of one component in the mixture
P	Precision
q	Number of kernels
q_c	Concentration of sample on location 2
r	Layers of pixels
R	Recall
W_e	Sample weight
W_{ab}^L	Weights and biases for connection ab in layer L
x	Input signal
x_i	Concentration in sample i
$\bar{x}$	Average of sample
$\hat{y}$	Prediction

List of abbreviations

AI	Artificial intelligence
AP	Average precision
AR	Average recall
ADE20K	Adela Barriuso 20 thousand images
CAM	Class activation maps
CNN	Convolutional Neural Network
CO_2	Carbon dioxide
EL	Elongation
ECDF	Empirical distribution function
FC	Fully connected
FN	False negative
FP	False positive
FPN	Feature Pyramid Networks
HTC	Hybrid task cascade
IoU	Intersection over union
mAP	Mean average precision
mIoU	Mean intersection over union
MLP	Multi-Layer Perceptron
MRI	Magnetic resonance imaging
MS COCO	Microsoft Common Objects in Context

RBG	Red blue green
RoI	Region of interest
RPN	Region proposal network
SI	Segregation index
TN	True negative
TP	True positive

List of figures

List of tables

Contents

1. Introduction

1.1. Project background

Particle-based products and services are integral to diverse industries, ranging from agriculture, ceramics, and chemicals to energy, geological systems, mining, pharmaceuticals, plastics, pollution control, and powder metallurgy. These industries collectively contribute to significant production volumes, emphasizing the widespread use of particles across various applications. In 2019 alone, global plastic production reached approximately 459 million tons [1], usable iron ores extraction amounted to around 2.6 billion tons [2], and the extracted coal was around 7.05 billion tons [3]. The agricultural sector also significantly contributes, with various particulate products totalling around 7 billion tons [4]. These figures underscore the immense scale and diversity of particle-based products. In many industries dealing with particulate materials, these particles often coexist in mixtures comprising multiple components. Components, in the context of bulk solids or mixtures, refer to groups of particles with similar distinguishing properties, which may include material type or particle size. Even within a single-material bulk solid, distinctions based on particle size can lead to separate groups of larger and smaller particles.

In manufacturing processes involving multiple components, precise control over the mixture's composition is essential to meet product specifications. The quality of the final product relies on maintaining the desired particulate composition. However, segregation, a counteracting process that opposes the mixing of components, poses a significant challenge. Segregation disrupts the homogeneous state of a bulk solid, potentially resulting in defects in the properties of the end products. Figure 1 illustrates the impact of segregation on a mixture, showcasing the transition from a well-mixed state to a segregated state, with darker particles surfacing on the right.

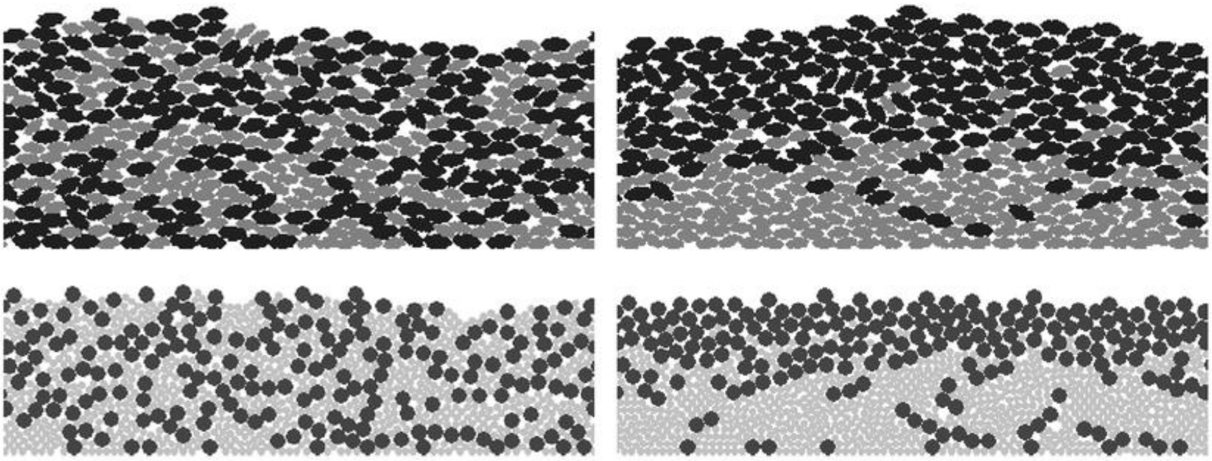


Figure 1 Segregation process, from mixed (left) to a segregated state (right) [5]

A well-known example of the industry for the effects of segregation is a blast furnace, where the bed permeability influences the efficiency of the process. Since gas is force through the bed which is the premise of the downward movement and reduction heating of the burden [6]. An efficient bed permeability is achieved by appropriately distributing the large and fine particles on the burden surface. Segregation of the burden materials affects the distribution of the materials in undesired ways [7]. Leading to negative effects on the permeability with inconsistent pressure drops over the materials which results in inefficient usage of the gas. Having consequences both economic and environmental [8]. The blast furnace, among several other cases in the industries that suffer from the phenomenon of segregation, highlights the importance of improving the understanding of the causes of segregation. Since segregation is a complex process influenced by numerous variables with diverse roots, there are lots of contributing factors to segregation. The influence of these factors is presented in various ways: the rate of segregation, the dominant segregation form, and the capacity to segregate [9]. Dependency on material properties and the environment further complicates the phenomenon. Differences in particle properties, including shape, morphology, elasticity, brittleness, density, chemical affinity, moisture absorbability, and magnetic properties, influence the segregation. Environmental influences, encompassing natural effects like humidity, wind, or gravity, and system-specific factors like surface roughness, vibrations, or transport modes, add to the intricacy of segregation.

To gain insights into segregation, experiments are conducted to recreate the process. Controlled environments in experimental setups help eliminate extraneous influences, allowing for the measurement and identification of segregation causes. Experiments often focus on variations in granular materials, such as identical particles, density differences, or size differences [10].

Despite segregation being a phenomenon that has been around for decades and is observed in common applications, extracting data on the composition of the granular mixtures is not a trivial task [11]. There are two distinct approaches for extracting segregation measurements, either through intrusive or non-intrusive methods. Intrusive methods involve extracting a sample of the mixture for detailed component analysis [12]. While sampling is straightforward, it destroys the structure of the mixture. Causing errors in its measurement and the following measurements [12]. On the other hand, non-intrusive methods employ a wide array of technologies, such as optics [13] or x-ray waves [14], to collect information without direct interaction with the mixture, preserving the undisturbed state for accurate measurements [12].

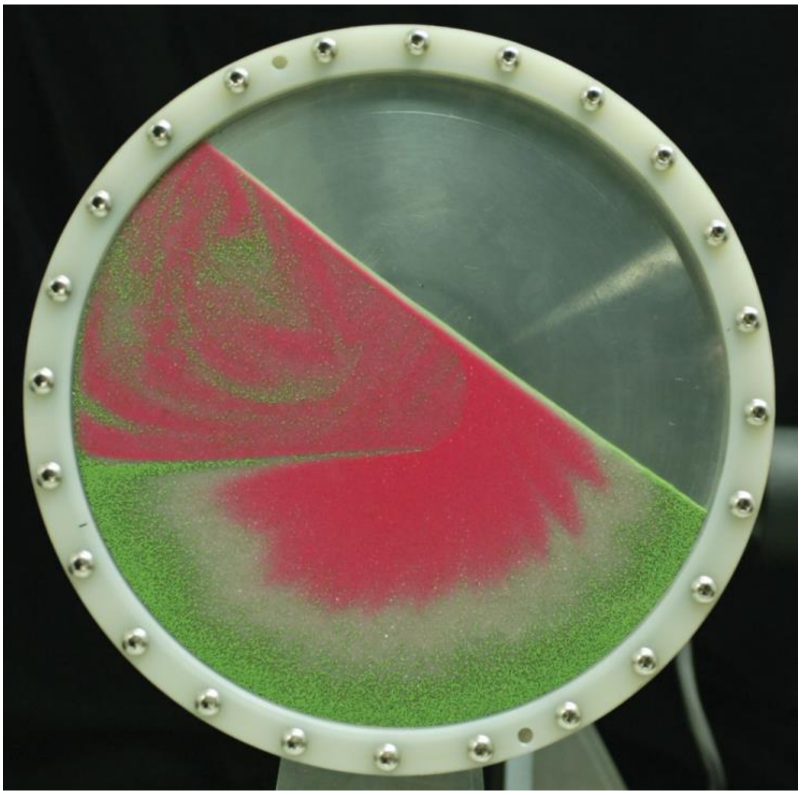


Figure 2 Three coloured component mixture in a rotary drum in front of a black background [15]

1.2. Problem definition

Optical measuring offers a cost-effective and straightforward approach to quantify segregation, rendering it highly appealing and suitable for application beyond specialized laboratory investigations [11]. However, its dependency on image processing for analysis necessitates conditions characterized by high colour contrast among particles and minimal light interference. This prerequisite imposes constraints on the choice of locations and the objects being measured, compelling the utilization of closed and specialized setups.

The attainment of adequate colour contrast may pose challenges when dealing with particles that exhibit similar colours, leading to difficulties in distinguishing them from each other. To mitigate this issue, a solution is to paint or coat particles to enable differentiation based on colour. In Figure 2, a purpose-designed setup illustrates a rotary drum with particles painted in three contrasting colours. The controlled lighting on the drum and the presence of a black background aim to minimize light disturbances, creating a distinct contrast between the particles and the background.

Given the constraints associated with the current optical measuring methods, conducting measurements in an open environment or with uncoloured particles is challenging. This limitation particularly hinders measurements in production lines or outdoors, especially when colouring particles is impractical. Recognizing these challenges, this thesis seeks to explore the potential of employing advanced algorithms, specifically artificial intelligence (AI). The objective is to leverage AI capabilities to facilitate segregation measurements in conditions where traditional image processing methods encounter difficulties.

1.3. Research questions

The main research question of this research is stated as:

- How can artificial intelligence be utilized to analyse and measure both material and particle size segregation in a granular mixture?

The main research question will be answered with the help of the following sub-research questions:

1. What is the state of the art for measuring and quantifying segregation in granular mixtures?
2. How can artificial intelligence be utilized through deep learning and computer vision to measure segregation in a granular mixture, and what are considerations for selecting a model and evaluating its performance?
3. How to select the most suitable artificial intelligence for measuring segregation in a granular mixture and what insights can the artificial intelligence provide on the composition and segregation of the mixture?

1.4. Scope

The objective of this thesis is to develop an artificial intelligence (AI) system with the capacity to discern distinct particles, enabling the measurement of segregation within a granular mixture.

The primary goals encompass the capability to identify particles and quantify segregation based on this recognition. Importantly, particle recognition should be achieved without the need for painting or coating, thereby ensuring applicability to materials or scenarios that prevent the need of particle preprocessing. Additionally, efforts are directed towards minimizing dependence on lighting, thereby removing constraints on measurement conditions and locations.

The particle recognition process should be adept at assessing the degree of segregation among materials within a mixture. The capacity for assessing segregation will be extended to encompass size segregation, relying upon the effectiveness of particle recognition.

The granular mixture used to identify particles and quantify segregation consists of the three materials commonly found in blast furnaces, coke, sinter and pellet. The recognition tasks are challenged by introducing three materials which have a lack of distinctive differences between them.

The anticipated outcome is the successful implementation of an AI system for particle recognition and segmentation, demonstrating proficiency in accurately identifying different particles and facilitating their conversion for segregation quantification. This system is expected to grant information concerning both material and size segregation within a granular mixture.

1.5. Approach

The methodology employed in this thesis involves an examination of artificial intelligence (AI) and its subcategories, including machine learning and deep learning, for the recognition of particles. Both machine learning and deep learning exhibit the ability to learn and detect patterns. Considering the characteristics of particle groups, three models will be selected for a comparative performance analysis. Opting for deep learning over machine learning was deemed more suitable for this thesis due to the former's inherent capacity to perform recognition without feature engineering and its enhanced generalization capabilities. Performance evaluation will be conducted using the F-score, a metric that balances precision and recall in a weighted harmonic mean. To assess a model effectively, a comprehensive setup is essential. This entails compiling a collection of images along with corresponding annotations, enabling the model to learn particle recognition. Post-learning, the model is presented with a new image to evaluate its performance.

The images utilized in this study make use of a mixture arranged in a heap and captured from an overhead perspective. Emphasizing the independence from lighting and particle colouring, none of the particles were painted or coated, and no additional lighting arrangements were employed. The images were captured using the camera of a smartphone, an iPhone 13 Pro.

The annotations encompass points outlining each particle, accompanied by a label indicating the material. These annotations serve as correct solutions during the learning process, enabling the model to learn both particle identification and material classification.

The learning process necessitates two sets of images: one for learning and another for monitoring progress. It is crucial that these sets do not share overlapping images to ensure that the model's performance reflects actual capabilities rather than mere reproduction of examples encountered during training. The sets comprise 180 images for learning and 35 for performance evaluation, resulting in a dataset of 215 images.

The expected output manifests as pixel markings corresponding to particles, labelled with the respective particle materials. Consequently, segregation indices akin to those employed in image analysis techniques can be defined. This involves creating a raster overlay on the image and comparing the distribution within sections to quantify segregation based on observed differences.

1.6. Report structure

The thesis is structured into six chapters, each addressing the research questions presented in the introduction. Chapter 2 initiates the discussion by exploring the phenomenon of segregation, introducing indices employed for quantifying segregation, and presenting the current state-of-the-art methods for measuring segregation in a mixture. Following this, Chapter 3 delves into the trade-off between machine and deep learning and provides an introduction to the workings of artificial intelligence (AI). The chapter also assesses suitable AIs for the task of measuring segregation, with a selection based on their performance using indices. Moving forward, Chapter 4 applies the chosen AI to the case of granular materials, primarily focusing on measurement procedures. It explores the AI's capacity to characterize the mixture using segregation indices and other pertinent data obtainable from the AI. Chapter 5 explores potential follow-up steps or ideas for expanding the range of information attainable. Finally, Chapter 6 concludes the research by addressing the research questions and summarizing key findings.

2. State of art for segregation measurements in granular materials

Before delving into measurement methods, it is important to establish an understanding of the phenomenon of segregation and its intricate nature. This chapter explores the appearances of segregation, its effects, and the particle properties that influence the process. Once this foundational understanding is established, subsequent sections will continue into the quantification of segregation through the examination of indices used to describe segregation in mixtures. Simultaneously, the chapter will scrutinize the state of the art methods employed to measure segregation in particle mixtures.

2.1. Introduction to segregation

Segregation denotes the process of separation between components or the demixing of a mixture [9]. A component, in this context, includes entities separable based on various properties such as size, material, or density. While the focus of this chapter is on the segregation of granular materials, it is important to note that this phenomenon extends beyond granular materials and can manifest in various mixtures, including gases, liquids, and different forms of solids such as powders.

In industrial processes requiring specific component ratios, materials are added and mixed to attain homogeneity, ensuring uniform products. However, segregation can disrupt this homogeneity, leading to sections within the mixture where a single component dominates or is nearly absent. This segregation can result in undesirable properties in the end products, with significant implications, especially in industries like pharmaceuticals, where it can be a matter of effectiveness versus lethality. Monitoring or controlling segregation is essential to mitigate the risk of producing malfunctioning products. When considering the mixture of coke, sinter and pellet, segregation greatly reduces the efficiency of a blast furnace [7]. As the segregation changes the particle size distribution over the burden, the gas flow is hindered due to irregular passages through the burden. The distribution of the gas is an important variable for the heat and mass transfer in the blast furnace [16]. Additionally, the blast furnaces for iron or steel production make up 70% of the energy consumption [17] and nearly 90% of all CO_2 emissions [18] of the entire production process. Therefore, it is crucial to optimize the operations of the blast furnaces to achieve better energy-efficiency and lower the carbon production by reducing the segregation.

Controlling segregation is challenging due to its diverse appearances and the ease with which it initiates. Five primary segregation mechanisms are recognized: percolation, flotation, transport, elutriation, and agglomeration [19] [20] [21].

- Percolation involves small particles filling gaps between larger particles, often encountered when filling containers or creating heaps.
- Flotation occurs when larger particles are pushed upwards by smaller particles, typically induced by vibrations.
- Transport happens due to the inertia of particles, where particles of various sizes or densities follow distinct trajectories.
- Elutriation is observed when small and light particles stay suspended in the air longer than coarse particles, commonly seen in finer powders.
- Agglomeration refers to the lumping together of particles, influenced by numerous factors such as humidity, heat, and pressure.

These mechanisms lack a common cause, making it challenging to address them collectively. Additionally, sensitivity to triggering segregation varies based on the materials and proportions within the mixture.

2.2. Method for measuring segregation in granular materials

In the context of segregating mixtures, the measurements must effectively reveal the composition of the mixture. The composition can be expressed in various ways, often in terms of concentration, representing the percentage of a component within the measurement. This concentration is determined based on the volume of particles. In the case of particles with distinct densities, mass can also be employed to determine concentration. Segregation for granular materials is commonly quantified based on either the particle size or material. These measurements are obtained through either invasive or non-invasive methods [22].

Invasive measurements involve influencing the mixture to obtain information, with one prevalent method being the extraction of samples. While the removal of a sample allows for more freedom in measuring and obtaining information, it introduces errors. Tools used to obtain samples can cause disturbances within the mixture, potentially inducing movements and segregation. Additionally, if the sample is not from the first or top layer, it may contain additional particles from the mixture's path to the sample area [12].

Non-invasive methods, in contrast, avoid direct contact with the experiment, utilizing alternative means for data collection. Commonly employed non-invasive methods include the use of light, such as photography, as well as radio waves and magnetism found in techniques like MRI [23], radar [24], and near-infrared [25]. These methods allow information to be obtained without direct contact. Image processing analysis, another non-invasive technique, is widely used due to its simplicity and cost-effectiveness. Unlike more complex methods, image analysis is limited to examining the first layer of particles. Image processing analysis involves three steps [26]: image compression, where the memory requirements are reduced, image processing, where filters like Gaussian and Laplacian are commonly applied, and image analysis, where calculations are performed on the processed image. Gaussian filters are used to reduce noise at the expense of sharpness, while Laplacian filters are employed for image sharpening and edge detection. Gaussian and Laplacian are used in conjunction to enhance an image [27], Figure 3 visualizes the effects of the filters. K-means algorithms and thresholding are additional techniques for grouping pixels based on values, providing alternatives for segmenting an image. These methods can be applied to subtract information from images, aiding in the quantification of segregation and tracking particle motion [27].

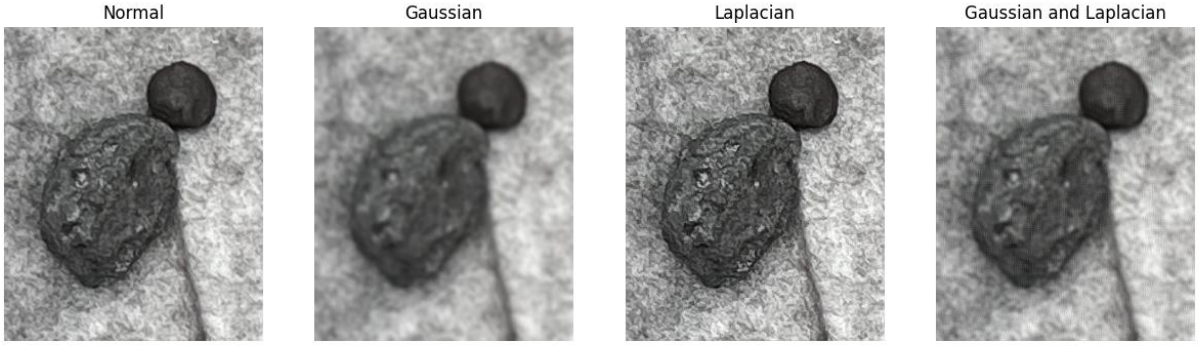


Figure 3 Applied effects of Gaussian and Laplacian filters

While image processing analysis has proven effective, it is essential to address challenges such as the potential similarity in colours of different materials and the need for stable lighting to minimize processing errors. Some particles may be painted or coated in a contrasting colour to aid in their separation from other groups, enhancing the distinction in the image. However, in practice, not all particles are capable of being painted. Stable lighting is crucial for consistent processing, minimizing shadows and reflections, and facilitating background correction to isolate particles [28]. Ensuring stable lighting requires an enclosure to be built around the setup similar to Figure 4. Limiting the intervention of foreign light sources with the measurements. Once image processing is complete, image analysis enables the segmentation of the image into different groups, allowing the determination of material concentration and the quantification of segregation [29].

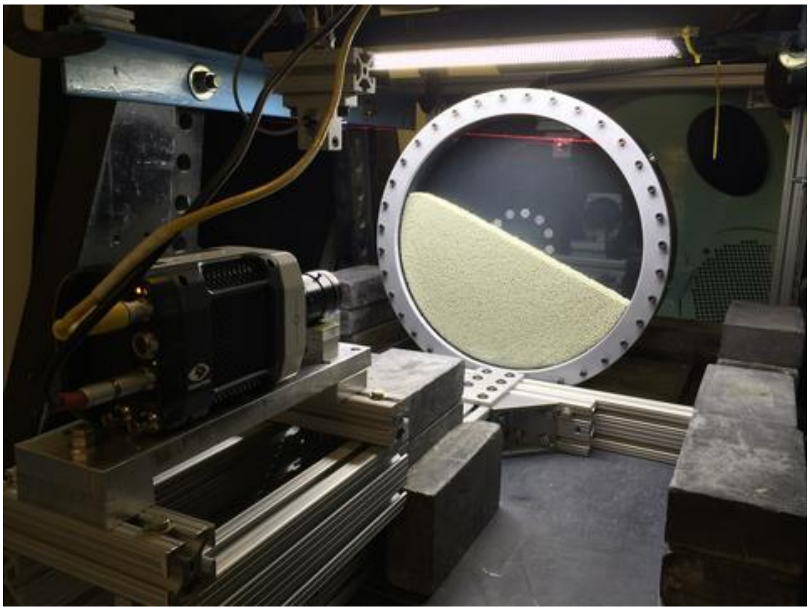


Figure 4 Enclosure for stable lighting on a rotary drum [30]

2.3. Quantifying segregation with indices

Once the particles in a measurement are recognized, a segregation index is used for quantifying the extent of segregation within a mixture. The measurements, derived from a sample, are compared to the intended proportions. The quantification of segregation involves assessing deviations from the intended values, often expressed as differences from the mean in cases where the designed ratios between components are unknown. Frequently used definitions within segregation indices are outlined in Table 1 [31].

Table 1 Frequent returning definitions for segregation indices

Name	Function
Average, sample arithmetic mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Sample variance	$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2$
Variance of a segregated system	${\sigma_0}^2 = P_c(1 - P_c)$
Variance of perfectly random system	${\sigma_R}^2 = \frac{P_c(1 - P_c)}{N_p}$

Variance is a common metric employed for quantifying segregation, especially in methods utilizing images or videos. It is noteworthy that for the estimation of the variance $\frac{1}{n-1}$ is preferred over $\frac{1}{n}$, $\frac{1}{n-1}$ aims to have an unbiased estimator. The segregation index, SI, is often quantified using equation (1), a widely accepted formula for this purpose, as evidenced by various studies [32] [33] [34] [35] [36] [37] [38]. As equation (1) only uses one component, it does not take other groups of particles into consideration. Therefore, allowing the application of the equation to mixtures with multiple components.

$$SI = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N - 1}} \tag{1}$$

Here, x represents the concentration within the sample, and N denotes the number of samples.
In contrast, when considering the counterpart of segregation, which is mixing, additional options for quantification come into play. The quantification for mixing characterizes the state of the mixture in terms of how well-mixed or segregated it is. The key distinction lies in the purpose of the measurement.

As the mixing index solely conveys the state of the mixture, the interpretation could vary. A well-mixed mixture exhibits minimal segregation, while a poorly mixed one displays more pronounced segregation. Mixing indices are categorized into four groups [31]: formula incorporating variances ${\sigma_0}^2$ and ${\sigma_R}^2$, formula that do not depend on the variances ${\sigma_0}^2$ and ${\sigma_R}^2$, formula for multicomponent systems and formula based on experimental work. Table 2 provides examples of mixing indices, drawn from the survey paper [31], which contains additional indices not explicitly mentioned. Regardless of the chosen index, the relationship ${\sigma_R}^2 > \sigma^2 > {\sigma_0}^2$ holds true for all cases. This is rooted in the understanding that a mixture's state can never be perfectly mixed or entirely segregated, and the variance can only fall between these two extremes.

Table 2 Mixing indices

Author	**Index**	$\boldsymbol{\sigma = \sigma_0}$	$\boldsymbol{\sigma = \sigma_R}$
Kramers [39]	$M = \frac{\sigma_0 - \sigma}{\sigma_0 - \sigma_R}$	0	1
Lacey [39]	$M = \frac{{\sigma_0}^2 - \sigma^2}{{\sigma_0}^2 - {\sigma_R}^2}$	0	1
Beaudry [40]	$M = \frac{\frac{\sigma_0}{\sigma} - 1}{\frac{\sigma_0}{\sigma_R} - 1}$	0	1
Valentin [41]	$M = \frac{\log \sigma_0 - \log \sigma}{\log \sigma_0 - \log \sigma_R}$	0	1
Sakaino [42]	$M = \frac{p_c\, q_c}{\sigma^2 W_e}$	P, q: concentration of sample W_e: sample weight	
Legatt [43]	$M = \frac{\sum (x_i - \bar{x})^2}{\bar{x}}$		
Lastovtsev [44]	$M = \sqrt{\frac{\sum (C_i - C_0)^2 n_i}{C_0^2 (n-1)}}$	C_0: concentration of sample for an ideal mixture C_i: concentration of sample i n: number of samples n_i: number of samples at concentration C_i	

2.4. Conclusion

In summary, segregation, a highly intricate phenomenon, is found across diverse industries, necessitating a comprehensive understanding and quantification. Measurement methods can be categorized as invasive or non-invasive, with the latter being preferred due to its minimal impact on the mixture's sensitivity. A non-intrusive method using optical measurements, particularly those utilizing cameras, is commonly employed. However, they face challenges in object recognition, requiring preprocessing techniques such as contrasting particle colours and controlled lighting setups. Which is only replicated in experimental setups, limiting the possibility of collecting data due to particles not being distinguishable. The literature proposes various indices, predominantly based on particle concentration, for quantifying segregation. Image analysis employs a comparable approach, focusing on concentration ratios derived from pixel sections.

3. Artificial intelligence for image processing

The fundamental principles of artificial intelligence important to vision systems as described in this chapter are derived from the book “Deep Learning for vision systems” by Mohamed Elgendy [45]. In this work, Elgendy explains the utilization of neural networks for the development of end-to-end computer vision applications, while providing in-depth insights aimed at equipping the reader with the knowledge necessary to comprehend research papers and interpret advancements in this domain. Consequently, this book serves as a primary reference throughout this chapter.

Artificial intelligence (AI) is a diverse area of computer science that concentrates on developing intelligent systems and algorithms, aiming to perform tasks typically carried out by humans due to the inherent decision-making involved. The goal of AI is to emulate human-like thinking, reasoning, problem-solving, and decision-making processes through the utilization of algorithms. These algorithms empower machines to process and comprehend information, make decisions, and learn from their actions, emulating human cognitive functions. The application of AI extends to exploring how computers can interact with the world and the various domains in which these technologies find utility [46].

In the realm of AI research and development, a spectrum of techniques is employed, encompassing machine learning, deep learning, computer vision, natural language processing, recommender systems, and robotics [47]. Machine learning, a subset of AI, focuses on creating algorithms capable of learning from data rather than relying on explicitly programmed rules [48]. This approach finds applications in diverse fields such as image recognition, natural language processing, and predictive analytics [47].
Deep learning, a subfield of machine learning, deals with artificial neural networks comprising a higher number of hidden layers, hence the term "deep" learning. These networks are designed to learn and extract patterns and features from data, demonstrating exceptional performance in tasks like image and speech recognition, natural language processing, and complex data analysis [49].
Computer vision, another AI subfield, revolves around teaching computers to interpret and understand their environment using visual inputs, commonly in the form of images and videos. Tasks like object recognition enable machines to "see" and comprehend their surroundings.

In the context of particle recognition, computer vision is employed to work with visual inputs, allowing for measurements with a camera. This accessibility makes the proposed measuring method viable for a wide user base, as most devices, including smartphones, are equipped with cameras. Given the inherent variability in experiments, particles, and images, predetermined algorithms are insufficient for information retrieval. Hence, particle recognition incorporates machine learning methodologies.

3.1. Trade-offs between machine learning and deep learning

In the realm of machine learning, an essential decision arises whether to adhere to conventional machine learning or delve into the subset known as deep learning. The primary distinction between these approaches lies in the neural networks they employ. While both machine learning and deep learning utilize neural networks, the latter employs deep neural networks, distinguished by a substantial number of hidden layers. The term "deep" in deep learning signifies the inclusion of neural networks with more than a thousand hidden layers, a significant contrast to the two or three hidden layers common in machine learning [50]. The alternative label for deep learning is “deep machine learning” referring to the large number of hidden layers.

The choice between machine learning and deep learning depends significantly on the application, as each option possesses distinct advantages and applications. A critical divergence lies in how they handle object features, which are informative patterns describing an object. Machine learning provides explicit control over features, allowing for manual crafting, often yielding quicker results. Conversely, deep learning can directly learn features from raw data, eliminating the need for handcrafted features. This autonomy enables deep learning to discern complex patterns and features [51], crucial in scenarios with intricate patterns, such as the variance among nearly identical particles in this study.
In the context of the amount of data, referring to the dataset employed for model training, machine learning demonstrates proficiency with limited labelled data. On the contrary, deep learning thrives on larger datasets to achieve robust generalization. Making the performance comparing similar to Figure 5. When faced with constraints in obtaining labelled data, machine learning might outperform deep learning [52].

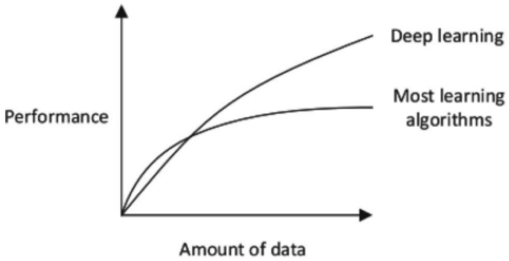


Figure 5 General performance comparison between deep learning and other machine learning algorithms based on data size [52]

Examining the performance of machine learning and deep learning, the latter exhibits the potential to attain state-of-the-art performance, characterized by a high level of generalization during training. This results in enhanced resilience to disturbances and elevated accuracy. However, the training duration for deep learning, compared to machine learning, is relatively prolonged and is dependent upon available computational resources, presenting a trade-off between performance and resource-time considerations [51].

Deep learning proves advantageous in image-related tasks, accommodating differences in object appearance across various scales and orientations. It can handle large numbers and diverse objects within a single model. In contrast, machine learning prefers consistent object appearances, necessitating preprocessing steps for input generalization to minimize variance. [53] [45]

An inherent dissimilarity exists in understanding the reasoning behind detection choices between machine and deep learning. Machine learning models permit the tracing of decisions through features, each contributing to a probability that collectively informs the decision. Deep learning, however, lacks known features and traverses numerous neural network layers before decision-making, making detailed scrutiny challenging. Yet, techniques like class activation maps (CAM) in object detection provide insight into contributing areas for decision-making [54].

For the application of detecting and localizing coke, sinter, and pellet particles, the concern lies in the limited distinguishing features among the particles. Features such as the round shape of pellets or the copper colour of sinter are subject to challenges such as partial coverage or variations in lighting conditions. The size of pellets is close to constant, while sinter exhibits significant size variability. Overlapping characteristics make feature crafting for machine learning become an intricate process due to lacking consistently discriminating features. The complexity intensifies as no singular characteristic can reliably separate the particles. [55]

To facilitate the measurement of segregation under varying conditions, including lighting, image size, and setup, flexibility is necessary. Additionally, variation in conditions causes the materials under consideration to have inconsistent visual representation in the images. The need for flexibility in handling diverse environments and the potential for state-of-the-art performance align with the decision to employ deep learning for the measurement setup. Moreover, deep learning accommodates the inclusion of additional particles in the dataset, potentially expanding the range of detectable particles and enabling broader applicability to bulk solids with fewer learned particles.

3.2. The usage of neural networks

The depth of the neural network is the main differentiator between machine learning and deep learning and most of the learning capabilities are in the neural network. Therefore, a brief overview is provided for the general workings of a neural network. A neural network is an interconnected system of numerous neurons and their respective connections. Analogous to the human brain, artificial neural networks seek to emulate this structure. Comprising layers of neurons or nodes interconnected by edges, an artificial neural network typically includes an input layer, followed by hidden layers, and culminating in an output layer. This architectural arrangement, as illustrated in Figure 6, is referred to as a multilayer perceptron due to its incorporation of multiple layers of nodes or perceptrons. The functioning of a perceptron, similar to neurons in the brain, involves the reception of signals, their processing, and subsequent transmission to the next perceptron contingent upon surpassing a specified threshold. The decision or prediction made by the model is influenced by how the signal undergoes processing and transmission.

To signify the significance of a signal, each input connection is assigned a weight value. The weight value, when high, imparts a more substantial impact on the decision-making process, amplifying the input signal. Conversely, low weights attenuate the input signal. In neural network representations, these weights are portrayed by the edges or connections from the input node to the perceptron.

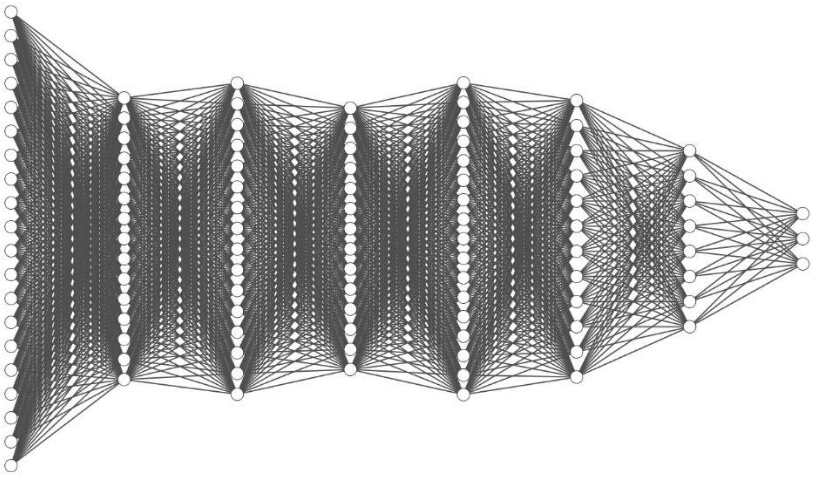


Figure 6 Multilayer perceptron

Internally, within a perceptron, the input signals undergo multiplication by their corresponding weights to yield a linear combination known as the weighted sum. The inclusion of a bias in the weighted sum allows for the linear combination to be shifted. Analogous to a linear equation, such as y = ax + b, where 'a' determines the slope angle, and 'b' introduces a shift along the y-axis, the bias in the weighted sum serves a similar purpose. This flexibility aids in better aligning predictions with the data.
The learning process of a perceptron involves iteratively adjusting the weights until the error between the prediction and the ground truth approaches zero. This process, referred to as the feedforward process, encompasses calculating the weighted sum and applying an activation function to make predictions. Subsequent adjustments to the weights are made to refine predictions in response to excessively high or low predictions. This iterative refinement continues until the error converges to a negligible value. Minimizing the error indicates close proximity of the predictions to the correct values, and the learned weights can then be preserved for future cases to facilitate predictions.
Termed as a perceptron, this neural network unit can predict the class to which an input belongs. The size of a perceptron varies, ranging from the minimal configuration comprising a single node to extensive networks housing millions of nodes. Larger neural networks are structured in layers, including input, hidden, and output layers. The input layer accepts information, the hidden layers make predictions based on weights and activation functions in nodes, and the output layer produces the final prediction. A simplified node is depicted in Figure 7, and the exemplary illustration in Figure 6 explains how these nodes can be expressed as equations to describe a network.

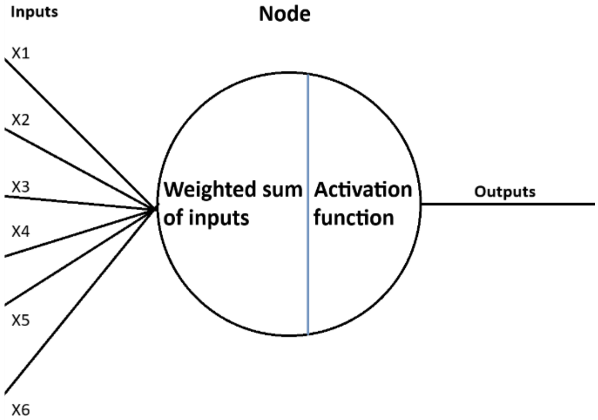


Figure 7 A single node with 6 inputs

The process involving the computation of a linear combination and the subsequent application of the activation function is termed feedforward. The designation feedforward conveys the forward progression of information from the input layer through the hidden layer to the output layer. This sequence unfolds through the execution of two successive functions: the weighted sum and the activation sum. The forward pass denotes the calculation across these layers to formulate predictions.

Each node encompasses a summation of inputs and an activation function, which dictates how inputs should be transmitted based on the chosen function. Frequently operating as a threshold mechanism, the activation function filters out inadequate values. Also referred to as transfer functions or nonlinearities, activation functions transform linear combinations of the weighted sum into a nonlinear model. Their incorporation introduces nonlinearity into the network, preventing a multilayer perceptron from functioning identically to a single perceptron, regardless of the number of layers. Additionally, these functions constrain output values to finite ranges.

Despite the surplus of activation functions developed in recent years, only a limited subset constitutes the predominant choices for practical application. Figure 8 illustrates some common types of activation functions, showcasing the diversity and variations within this essential component of neural network architectures.

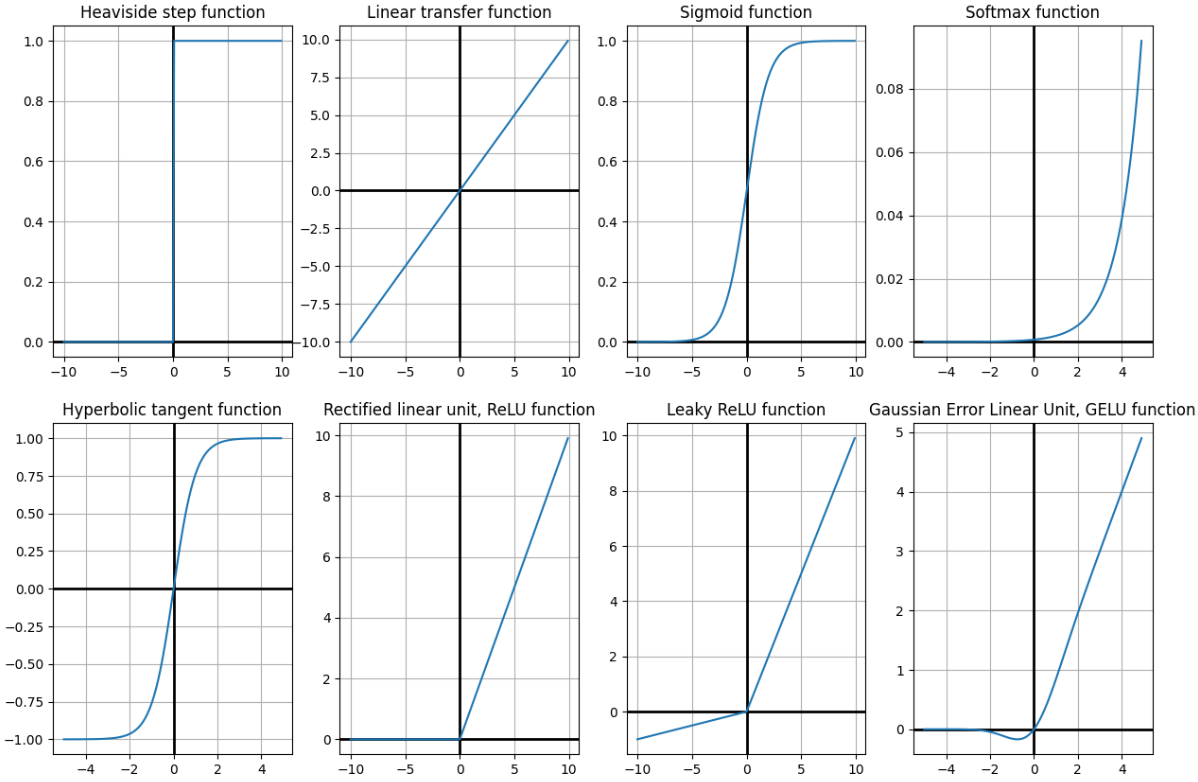


Figure 8 Activation functions

To exemplify the computations involved in the feedforward process, we will consider the neural network depicted in Figure 9. This particular example comprises 3 inputs and 4 layers, with node configurations of 3, 3, 4, and 1 for each layer, respectively. The notation W_{ab}^{L} is used signify the weights and biases, where 'L' signifies the layer number, 'ab' designates the weighted edge connecting neuron 'a' in layer 'L' to neuron 'b' from the previous layer.

For the activation function σ(x) is used to represent a sigmoid activation function. The nodes are represented as a_m^L where 'L' is the layer and 'm' is the node index in the layer.

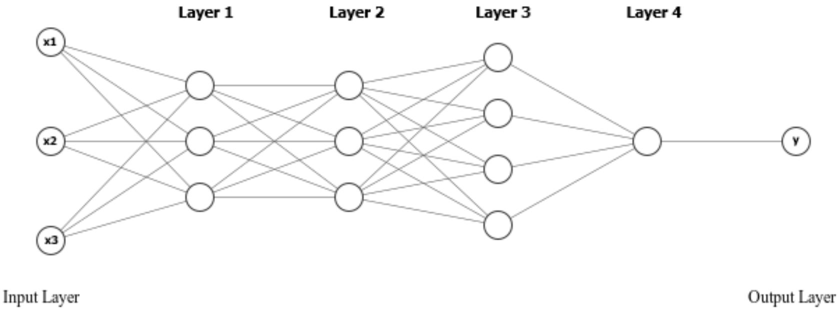


Figure 9 Neural network with 4 layers, 3 inputs and 1 output

Writing Figure 9 in equations form the prediction:

$$a_1^1 = \sigma(W_{11}^1 * x_1 + W_{12}^1 * x_2 + W_{13}^1 * x_3)$$
$$a_2^1 = \sigma(W_{21}^1 * x_1 + W_{22}^1 * x_2 + W_{23}^1 * x_3)$$
$$a_3^1 = \sigma(W_{31}^1 * x_1 + W_{32}^1 * x_2 + W_{33}^1 * x_3)$$

The second and third layers have similar equations for a_1^2, a_2^2, a_3^2, a_1^3, a_2^3, a_3^3 and a_4^3 all the way to the output prediction $\hat{y}$ in layer 4:

$$\hat{y} = a_1^4 = \sigma(W_{11}^4 * a_1^3 + W_{12}^4 * a_2^3 + W_{13}^4 * a_3^3 + W_{14}^4 * a_4^3)$$

This representation is for a simplified four-layer neural network with three inputs. The complexity and number of equations significantly increase when working with a higher number of nodes in the input or layers.

A more efficient approach for prediction computation involves using matrices. This not only enhances the visibility of equations but also accelerates the computation process. For the given example network, the prediction in matrix form is expressed as:

$$\hat{y} = \sigma * W^4 * \sigma * W^3 * \sigma * W^2 * \sigma * W^1 * x$$

$$\hat{y} = \sigma[W_{11}^4 \quad W_{12}^4 \quad W_{13}^4 \quad W_{14}^4] * \sigma \begin{bmatrix} W_{11}^3 & W_{12}^3 & W_{13}^3 \\ W_{21}^3 & W_{22}^3 & W_{23}^3 \\ W_{31}^3 & W_{32}^3 & W_{33}^3 \\ W_{41}^3 & W_{42}^3 & W_{43}^3 \end{bmatrix} * \sigma \begin{bmatrix} W_{11}^2 & W_{12}^2 & W_{13}^2 \\ W_{21}^2 & W_{22}^2 & W_{23}^2 \\ W_{31}^2 & W_{32}^2 & W_{33}^2 \end{bmatrix} * \sigma \begin{bmatrix} W_{11}^1 & W_{12}^1 & W_{13}^1 \\ W_{21}^1 & W_{22}^1 & W_{23}^1 \\ W_{31}^1 & W_{32}^1 & W_{33}^1 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The output prediction $\hat{y}$ is made within a range of zero and one by applying for example the SoftMax activation function. The output prediction is made into a probability distribution, providing the probability for all unique outputs. In the example of Figure 9, the output would result in the probability the input belongs to the singular class. For networks with more output classes, a node is required per class. [53] [45]

During the training phase, the weights and biases of the prediction equation undergo iterative adjustments. A loss function is employed to gauge the magnitude of the error by comparing the prediction against the expected output. The objective is to minimize the error, as a value close to zero indicates a more accurate prediction. Various loss functions can be used, with mean squared error and cross-entropy being common examples. The mean squared error considers the squared difference between the expected and actual values, averaging the result. This ensures a positive error value, with a smaller error indicating a higher likelihood of correct predictions. In contrast, cross-entropy compares distributions. Applied to machine learning predictions, this loss function involves taking the natural logarithm of the product of predicted and expected values, resulting in a value between zero and one. A smaller value signifies improved performance.

The training process utilizes the gradient descent algorithm [56], an optimizer designed to approach optimal performance. This algorithm manipulates the weights and biases within the network based on initial random values for predictions. The error is determined using metrics like mean squared error, and adjustments to weights and biases are made accordingly. The size and direction of these changes are influenced by the error magnitude and location within the error graph, with more significant adjustments made for larger errors. The training rate variable scales the size of these changes before applying them to the chosen weights and biases.

Back propagation, a key process within gradient descent, involves determining the changes for the weights. This process extracts information on predicted results, total error, and weights to update the weights for more accurate predictions. While the optimizer suggests that repeated iterations can lead to near-perfect performance, excessive training may cause the model to interpret irrelevant information, associating decisions with noise in the image. This overfitting phenomenon results in exceptional performance on training data but a significant drop in performance on unseen images. To prevent overfitting, training sessions may be halted earlier, or additional data may be introduced. Conversely, underfitting, the opposite problem, occurs when the model has not been trained sufficiently to recognize patterns effectively [57]. Striking a balance during the training process is crucial to ensure the model generalizes well to new data.

To assess performance and track progress during training, validation cycles are incorporated between iterations. Validation involves exposing the model to new data to measure its performance, allowing comparison with training results. Fluctuations or a decline in performance within the validation process, alongside a decreasing loss function, may indicate overfitting. Conversely, a diminishing loss function and an increase in performance suggest underfitting.

Finally, once training concludes, test data is employed to evaluate and visually inspect the model's performance. While evaluation yields numerical indicators of performance, test data offers a tangible assessment of the model's capabilities.

3.3. Computer vision

In the context of quantifying segregation in granular materials, the emphasis is on the domain of computer vision, a field that involves providing artificial intelligence (AI) with inputs to interpret the environment and respond accordingly [58]. Computer vision primarily deals with visual inputs such as images and videos, encompassing tools like Lidar or radar for generating depth maps based on reflection points. This capability is often harnessed for tasks like monitoring areas or processes [59], automating mundane tasks without human intervention, and responding dynamically to the environment in applications like autonomous vehicles [60] and robotics.

Initially, computer vision was equated with image processing analysis. However, contemporary perspectives recognize image processing as just one facet of the broader and intricate systems dedicated to interpreting the content within images. Machine learning applications for computer vision frequently integrate various image processing features [61], exemplified by colour-based detection, similar to thresholding but tailored to specific colours. On the contrary, deep learning models may integrate image processing tools within feature extraction pyramids, but they do not depend on these tools. Deep learning models can discern patterns without the need for separate tools, creating case-specific filters for application in convolutional layers, a concept to be explored further in the subsequent section 3.4.1.

Common to diverse computer vision tasks is the fundamental query: "What are we looking at?" This question underpins three distinct recognition levels built upon each other. The first level involves image classification, where an image is labelled based on the detected elements within it. This task provides considerable flexibility in terms of the detection targets and corresponding labels, such as identifying animals or defects in a production line. A more intricate task follows, combining object identification and localization, wherein each recognized object is labelled, and a bounding box is assigned to delineate its spatial extent. This bounding box facilitates contextual understanding and action based on object placement, a significant advancement in robotics. Further enhancement in object identification includes landmark detection or keypoints, which identifies crucial features within objects, commonly applied for tracking human motion in activities such as sports [62].

Building on object identification and localization, the final recognition level is segmentation. Segmentation involves marking pixels corresponding to identified objects and creating a detailed representation of their shapes and locations. This process results in binary values stored as masks, with semantic segmentation assigning a class to pixels belonging to an object group, while instance segmentation labels pixels per individual object. This nuanced understanding facilitates precise targeting of specific object pixels rather than entire groups.

The training of deep learning models is categorized based on the learning method, broadly classified into supervised [63], unsupervised [64], hybrid [64], and reinforcement learning [65]. Supervised learning, the most prevalent method, necessitates the provision of solutions for the learning process, the model learns by minimizing the error between its predictions and the provided solutions within a dataset. Unsupervised learning, conversely, identifies correlating patterns within large datasets, often employed in generative models like DALL-E [66]. Reinforcement learning is applied when models interact with the environment, optimizing based on reward policies that score model performance, and fostering improvement through positive reinforcement. Hybrid models seek to combine aspects of various learning methods to utilize their respective strengths.

In the specific context of measuring segregation, the approach involves segmentation trained through supervised learning. Unsupervised learning encounters challenges due to the absence of supervision [67], while reinforced learning, although emerging for image segmentation, necessitates a comparable loop to supervised learning for performance evaluation against annotated datasets [68].

3.4. Deep learning architecture components for computer vision

Before venturing into the performance comparison of various models, the structural components that constitute these models are examined. This exploration aims to raise an understanding of the model's elements, which will guide the selection process for evaluating their performance. The insight on the components and performance within this chapter is solely derived from the literature. Moreover, the focus here is exclusively on models relevant to computer vision and segmentation tasks.

At a high level, a model presents a simplified structure that outlines its distinctive features and the specific problem it addresses. Among the array of available networks, Convolutional Neural Network (CNN) stands out for excelling in tasks related to image data and classification, prerequisites for segmentation tasks [65]. Tailored for processing structured arrays of data, such as images, CNNs have established themselves as the state-of-the-art solution for numerous visual applications in computer vision. Notably, CNNs demonstrate a robust capacity for recognizing intricate patterns in images, encompassing elements like lines, circles, and even more complex patterns such as facial features.

Structurally, CNNs share similarities with Multi-Layer Perceptron’s (MLPs), featuring multiple convolutional layers, followed by activation and pooling layers, culminating in fully connected (FC) layers at the end. The convolutional layers within a CNN exhibit the remarkable capability to discern sophisticated patterns. For instance, a network comprising three or four convolutional layers could proficiently recognize handwritten digits, while a more complex architecture with 25 layers could distinguish human faces.

When referring to the "backbone" of a model, we are referring to networks like CNNs. In some instances, the primary distinction between models lies in the specific version or variant of CNN that is employed.

3.4.1. Convolutional layers

Convolutional layers serve as crucial components within convolutional neural networks (CNNs), functioning as feature extraction windows that navigate input pixels to identify relevant features crucial for object recognition. These convolutional kernels, conceptualized as square areas, systematically traverse an image, searching for distinctive patterns. A single layer incorporates numerous kernels, each adept at recognizing specific patterns. The convolutional operation yields feature maps, subsequently processed by an activation function to produce the final outcomes.

A kernel, depicted as a grid of discrete numerical values, possesses weights adjusted during the training process to discern significant features. The convolution operation involves multiplying these weights with corresponding pixels in the receptive field, resulting in a sum. This process, repeated across the entire input, generates convolved images known as feature maps or activation maps. Each kernel produces a distinct feature map within a convolutional layer [69].

Illustrating this process through an example, consider the application of a single kernel to a colour image. A colour image is characterized by three colours represented in the RGB space, with each colour ranging from 0 to 255. The convolutional layer's input is a three-dimensional matrix, reflecting the image's pixel dimensions and colour channels (m x n x r). Concurrently, kernels possess similar three-dimensional dimensions (p x p x q). The parameterization demands that the dimension p of the kernel must be less than both the width m and height n of the input image. Furthermore, the third dimension q of the kernel is not constrained to the depth r. As the kernels traverse the input, the resulting feature maps' dimensions are smaller than the input, impacting both width and height.

Figure 10 presents a starting image in its original colours together with the extracted three RBG colour layers. The image is used to illustrate the workings of a convolutional layer.

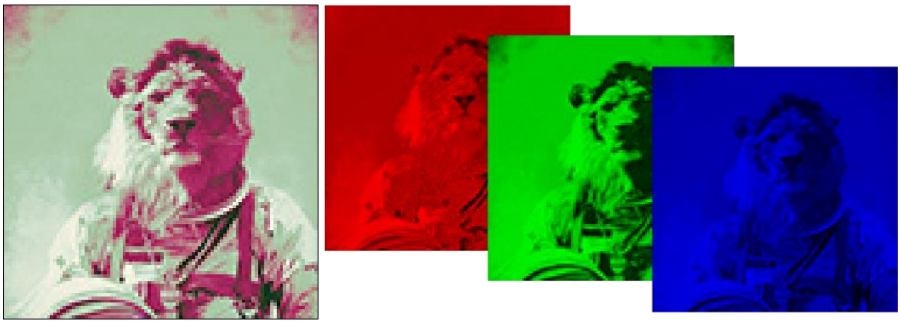


Figure 10 Image [70] and the extracted RBG layers

For demonstration, the focus is on the red layer of a colour image represented by a matrix in Table 3. Table 3 contains the colour intensity of the top left corner of the red layer. An edge-detection kernel, exemplified in Table 4, is applied to highlight edges based on intensity differences within the colour.

Table 3 Top left pixels of the red layer from Figure 10

189	187	165	193	182
181	191	178	183	181
184	193	188	185	187
192	188	187	192	179
170	173	190	193	183

Table 4 Kernel with edge detection filter

0	-1	0
-1	4	-1
0	-1	0

For the first iteration, the receptive field is highlighted, and the values from Table 3 are multiplied by the kernel. The sum is then stored in the convoluted image at the corresponding destination pixel. For the initial operation at the top-left position, the convoluted image's top-left pixel is determined. For subsequent operations, the kernel is shifted one column to the right to determine the next pixel of the output. The convoluted image has dimensions (m-p+1 x n-p+1) after all operations. In the given example, considering a 100 x 100 pixel image, the resulting size of the convoluted image would be 98 x 98.

Table 5 Image multiplied by the kernel to obtain a convoluted image

189	187	165	193	182
181	191	178	183	181
184	193	188	185	187
192	188	187	192	179
170	173	190	193	183

25			

The sum of the multiplication for the highlighted operation would be:

189x0 + 187x-1 + 165*0 + 181*-1 + 191*4 + 178*-1 + 184*0 + 193*-1 + 188*0 = 25

The convolution operation, detailed in Table 5, demonstrates how the kernel traverses the image, producing a convoluted image, Figure 11, emphasizing contours and shapes.

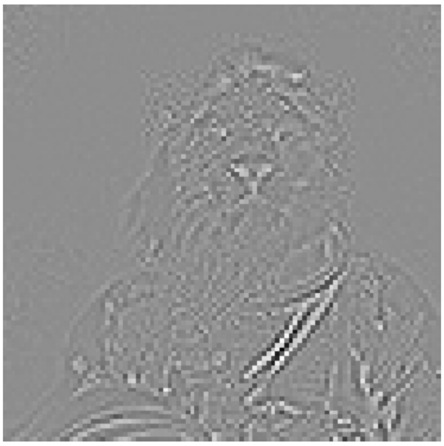


Figure 11 Convolution on the image from Figure 10

The design of convolutional layers involves decisions on the number and size of kernels. Theoretically, larger kernels with more weights offer enhanced learning potential but increase computational complexity. Common kernel sizes include 2x2, 3x3, 5x5, and 7x7, with occasional deviations like 9x9 or 11x11 [71]. The 1x1 kernel, however, serves a different purpose, intensifying non-linearity without decreasing the receptive field [72].

Additionally, the number of layers impacts a model's learning capacity and performance. An increase in layers generally improves feature learning [72] [73] but can lead to overfitting if not regulated effectively

[74]. The number of layers also affects the model's input size, limiting it to prevent convolution from reducing to a 1x1 output. Larger inputs are resized or cropped, potentially resulting in unintended information loss.

3.4.2. Pooling Layers

Similar to the convolutional layers, where each kernel has optimized weights, networks with multiple layers and kernels result in a considerable number of weights, leading to increased mathematical complexity during the learning process. Pooling layers address this issue by reducing the parameters transmitted to the subsequent layer. Through a pooling process that employs summary statistical functions like maximum or average, the input undergoes resizing, resulting in fewer parameters. This downsampling of the feature map from the convolutional layer diminishes computational complexity. Pooling layers are strategically positioned after every one or two convolutional layers.
In the pooling layer, akin to convolutional layers, a kernel traverses the convoluted image. However, the pooling kernel selects specific values to retain. Max pooling retains only the highest values, while average pooling computes the average of all values within the kernel. This summarization condenses a portion of the image into a single pixel, effectively reducing the feature map for subsequent layers.

To illustrate the impact on the feature map, the feature map from the convolutional layer undergoes downsampling in a pooling layer. The outcomes are depicted in Figure 12, utilizing a 2x2 kernel with a stride of 2. Stride, denoting the step size of the kernel, signifies the movement every two rows or two columns when traversing input values. In contrast, a stride of 1, as employed in the convolutional layer, advances to the next column or row. Additionally, padding can be applied to the pooled image, maintaining its size. Given multiple convolutional and pooling layers, each operation reduces the image size. To prevent the elimination of pixels, padding in the form of zeroes is added around the pooled image. For instance, a padding of 2 would introduce two rows above and below the pooled image and two columns on either side.

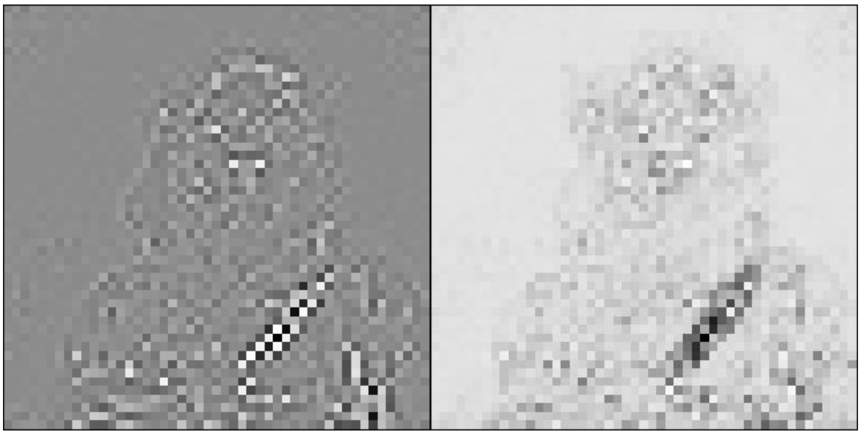


Figure 12 Downsampling, average pooling (left) and max pooling (right)

Similar to the convolutional layer's kernel size, there is no universal solution for the size of the pooling kernel. It becomes a network parameter subject to tuning based on preference. The same applies to stride and padding, serving as additional parameters to be adjusted during CNN design.

3.4.3. Notable additional features

Within the realm of CNN components, namely convolutional, activation, and pooling layers, there is flexibility for making up the structure. As observed with convolutional layers, there are no rigid guidelines to adhere to, allowing for the design of structures tailored to specific problems. Consequently, numerous slightly different versions of CNNs are available. Despite this flexibility, only a handful of distinct CNNs are commonly employed across various use cases. This section explores extensions applied to models or CNNs that enhance model performance, focusing on those deemed promising based on component evaluation and available literature. The analysis is confined to models and CNNs within the domains of deep learning and computer vision, specifically for instance segmentation tasks.

Two extensively researched and widely used extensions that enhance model prediction quality are the Feature Pyramid Network [75] and region of interest (RoI) Align [76]. Feature Pyramid Networks (FPN) address the challenge of feature recognition at multiple scales by generating feature maps with varied resolutions and levels of detail. This assists in extracting features more effectively from the convolutional neural network (CNN). FPN is integral to the feature extraction process, offering several options for passing on feature maps, as shown in Figure 13. These options include resizing the image into different sizes (image pyramid) and extracting different-sized feature maps (feature pyramid). Alternatively, convolutions can be utilized to create feature maps of varying sizes, with options to pass on only the smallest or last feature map or all feature maps. There is also the option to enrich feature maps with details from higher levels. This comprehensive approach, known as a Feature Pyramid Network, significantly enhances object detection and segmentation performance compared to using either nothing or an image pyramid [75].

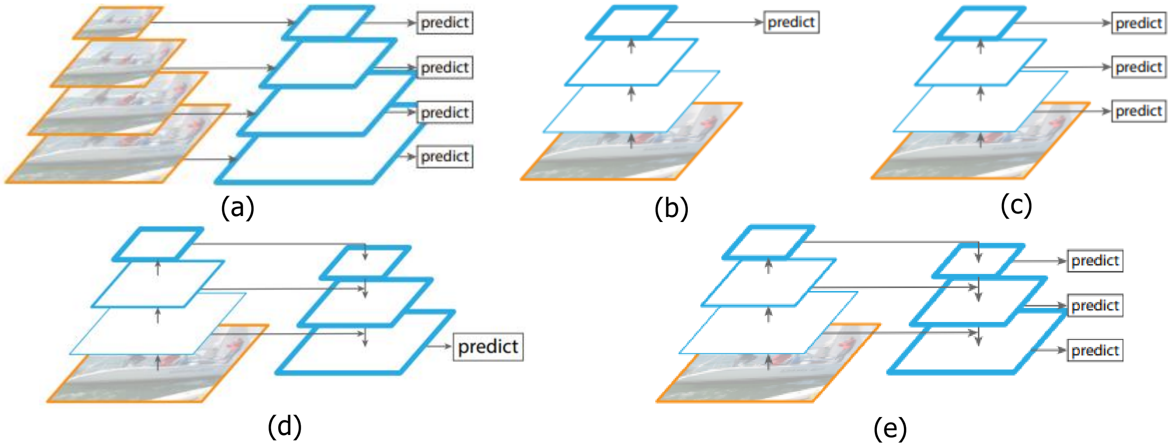


Figure 13 Image pyramids. (a) feature pyramid, (b) feature map, (c) multiple feature maps, (d) top-down feature map, (e) feature pyramid network [75]

RoI Align comes into play after a region proposal network, refining regions proposed by the network. Since the convolutional layers alter the size of feature maps from the original image, RoI Align proposes adjusting coordinates based on the number of convolutions the feature map undergoes. This adjustment ensures the new coordinates are relative to the feature map's size. To align the region of interest with the feature map's grid, RoI Align divides the coordinates, creating sections (bins). For each bin, RoI Align selects four points and utilizes bilinear interpolation to assign a specific value to these points. This process is followed by obtaining maximum or average values within the bins, similar to the pooling layer. RoI Align thus constructs the feature map for the proposed region of interest, which is then utilized for bounding box predictions, classification, and segmentation masking.
In addition to RoI Align, there are RoI Pooling and RoI Warp. However, due to differences in their approaches for determining values within bins, both RoI Pooling and RoI Warp are less preferable for segmentation. A comparison between the two, presented in the introduction of Mask R-CNN [76], illustrates a clear performance gap.

In more recent developments, deformable convolutional networks have garnered attention for making dense predictions, such as those required for tasks like semantic segmentation and depth map prediction [77]. Deformable convolutional networks introduce an additional step within convolutional layers, allowing the kernel's locations to have variable values. Unlike conventional convolutional layers, which derive values from around the kernel's central pixel, deformable convolutional layers incorporate an extra set of parameters representing offsets for each grid spot in the kernel. Similar offset additions are applied to RoI pooling. This approach enhances accuracy and localization capabilities, particularly for objects with inconsistent shapes [78].

Apart from traditional convolutional networks and deformable convolutional networks, transformers can serve as a backbone. While transformers are commonly applied in natural language processing models, recent exploration has extended their use to object detection and segmentation [79]. The transformer's approach differs from CNNs, dividing an image into sections that are linearly embedded to create a sequence of vectors representing the image [80]. The primary mechanism of a transformer is self-attention, emphasizing the importance of critical aspects in an image and focusing more on relevant features. Additionally, the self-attention mechanism aids in modelling dependencies between input sequence elements over large ranges, improving generalization capabilities. This is achieved by sparingly utilizing the neighbourhood structure, a departure from CNNs' common reliance on kernel operations [79]. While transformers can function as a standalone backbone, they can also collaborate with a CNN by linearly embedding the feature maps from the CNN and passing them into the transformer. This hybrid architecture structure achieves even higher accuracy [80].

FPN is easily implementable with any traditional backbone [75] and is integrated into deformable CNNs [78]. RoI Align can be applied to any architecture involving region proposal networks as an alternative to RoI pooling. Deformable CNNs relinquish the fixed orientation of the RoI kernel by applying offsets, contributing to increased accuracy and enhanced localization capabilities, particularly for objects with inconsistent shapes [78]. Given the distinct approaches of RoI Align and deformable CNN, a combination of both does not exist.

3.5. Considerations for potential image processing models

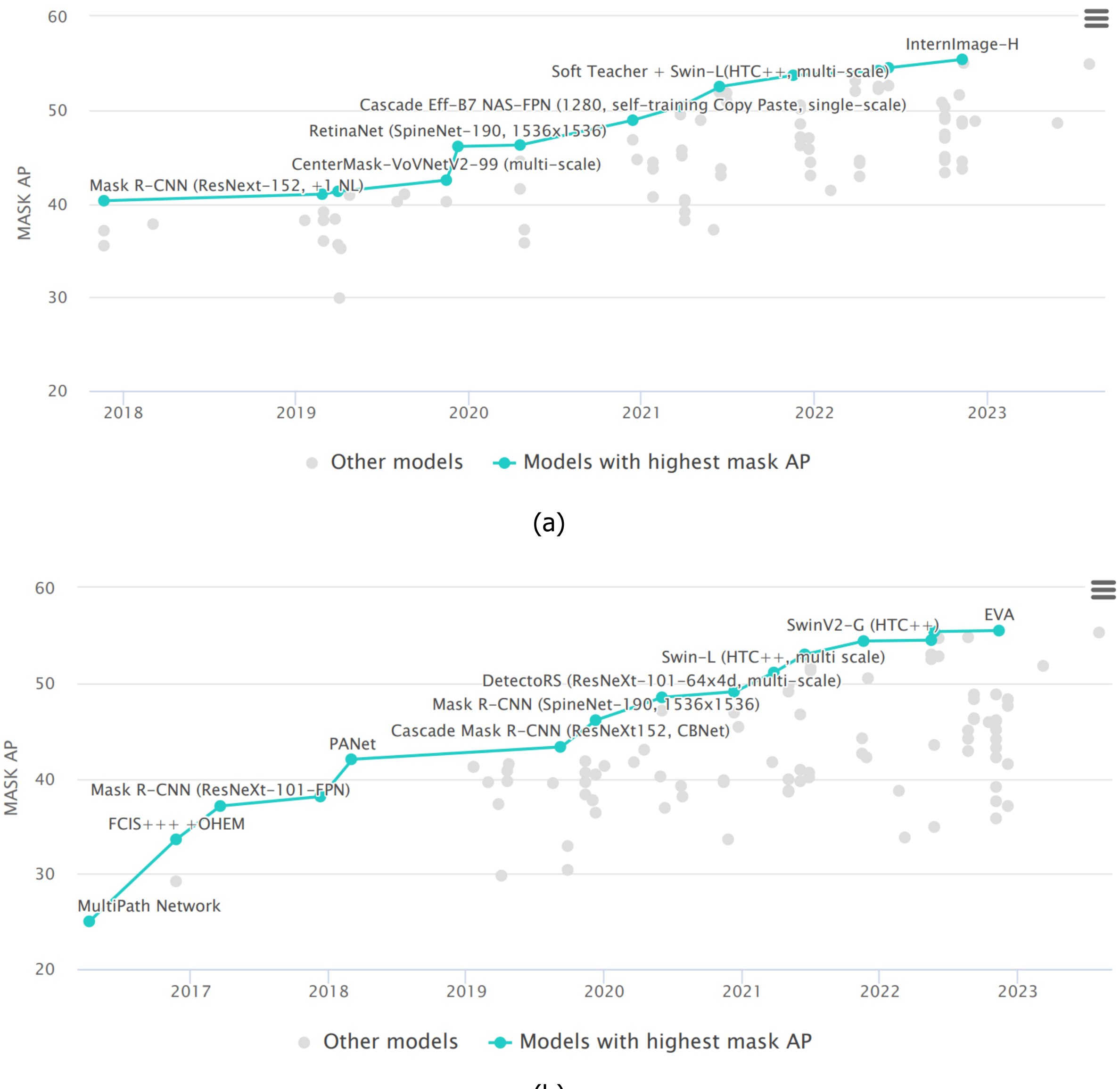
Amongst the extensive array of models, containing multiple iterations with alterations in the backbones, a selection of three models is introduced. These chosen models will undergo comprehensive training and subsequent performance comparison in the specific context of instance segmentation for granular materials.

In addition to evaluating models based on referenced features or methods embedded within their architectures, performance indicators gathered from relevant literature serve as benchmarks for the coming comparisons. As there is no definitive method of determining the performance capabilities, a comparison on the same dataset is done to provide insight. Therefore, important models are frequently subject to comparative analyses for establishing benchmarks, particularly employing datasets such as MS COCO [81], ImageNet [82], ADE20K [83], and Cityscapes [84]. These datasets, encompassing diverse image volumes with annotated objects across myriad categories, provide standardized metrics for assessing model performance. MS COCO, in particular, stands out as the predominant choice for benchmarking instance segmentation performance due to its realistic representation of common objects in natural environments. Within MS COCO there 80 different common objects labelled across 328k images.

However, it is crucial to acknowledge that performance metrics derived from standardized datasets may not universally translate to other datasets. Despite this limitation, such measurements offer insights into a model's learning capabilities. The distinguishable features and performance indicators from standardized datasets inform the strategic selection of potential models.

Drawing insights from benchmarks curated by Papers with Code [85], as depicted in Figure 14, enables informed speculation on models exhibiting high learning capabilities and prevalent trends. Notably, recent trends underscore a reliance on transformer-based models, evident in the top scores achieved over the past two years. Consequently, a detailed examination of available models, particularly those leveraging transformers, is warranted.

Upon exploring transformer-based models, hybrid models contemplate the integration of a transformer with a convolutional neural network (CNN) or a deformable CNN. When examining top-performing model groups that share similar backbones or structures, the potential for comparable performance becomes evident. Intriguingly, besides the shift toward transformers, models established on Mask R-CNN and Cascade Mask R-CNN [86] continue to be created. Their sustained presence, despite their initial release several years ago, underscores their enduring relevance in the field of artificial intelligence development. Noteworthy, is the substantial increase in Average Precision (AP, 3.6.3) exhibited by top-performing models, demonstrating an enhancement of ten to fifteen per cent over preceding iterations.



(a)

(b)

Figure 14 Performance graph of instance segmentation on (a) the COCO minival dataset and (b) the COCO test-dev dataset [72]

The models considered for the selection have to be capable of performing segmentation. For the purpose of measuring segregation, instance segmentation is required of the model. Instance segmentation allows for both material and size segregation measurements while semantic segmentation would only be able to perform material segregation measurements. Additionally, as the models develop at a rapid rate, many models are freely available. However, there are still models that are not made publicly accessible. The latter is often the case for newer or more nuanced models. Therefore, only models that are public are taken into consideration. The discussed FPN in section 3.4.3 is seen as a requirement for the models. However, as almost all models have an FPN included, this requirement does not affect the options. When presented with the option of selecting a backbone, as mentioned in section 3.4.1 the backbone with more parameters often allows for better performance and is therefore preferable. The criteria for a model to be considered are the potential or the popularity. These criteria are for models that comply with the task of instance segmentation. The potential of a model is based on its performance relative to other models, often shown in the form of performance scores on a dataset. The popularity of a model is seen by the frequent appearances in performance comparison tables. Alternatively, the models' popularity could also be represented by having models released based on them.

When considering instance segmentation models, the model Mask R-CNN has to be mentioned. Mask R-CNN is considered to be the most basic model for instance segmentation [87] and a classic [88]. Thus, it is not uncommon to see Mask R-CNN being referenced in comparison tables. To name some papers in which Mask R-CNN is used for comparison: MEInst [89], SOLOv2 [90], YOLACT [91] and CenterMask [92]. Therefore, Mask R-CNN is included in the performance comparison.

A survey paper is taken to limit the number of potential models. A survey paper filters the models that show limited innovation or are minor adjustments on other models out. Resulting in a more distinct and sizable group of models to narrow down a select few. Hence, the survey paper of Sharma [93] is taken to reduce the pool for the selection. To start with the performance potential of the models, Sharma presents a table with an overview of the performance on standardized datasets. From the table, the best performing model among the presented models is ISTR [94], a transformer based model. The potential of transformer models in the overview echoes the trend seen in Figure 14.

A popular technique that is presented in several high scoring models is cascade learning [95]. Cascade learning takes the outputs through several iterations to improve the prediction. A similar iterative approach is seen in transformer based models and hybrid task cascade models (HTC) [96]. The model to bring cascade learning to instance segmentation is Cascade Mask R-CNN [86] [96]. Besides the conventional approach of using a CNN, Cascade Mask R-CNN is also used with a transformer backbone such as Swin Transformer [97]. Cascade Mask R-CNN inspired the HTC branch of models that is seen in some of the best performing models, as Swin-L or SwinV2 in Figure 14 b. Since Cascade Mask R-CNN has such influence on the development of other models and the relevance of the model to date, Cascade Mask R-CNN is added to the group of models for the selection.

With a selection of three models that have shown their relevance or performance for image segmentation, the model selection includes Mask R-CNN [76], Cascade Mask R-CNN [86] and a transformer based model, ISTR. All three models incorporate a Feature Pyramid Network (FPN). While Mask R-CNN and ISTR utilize a Resnet101 backbone, Cascade Mask R-CNN employs a Resnet50 backbone. The distinction between Resnet50 and Resnet101 primarily lies in the number of convolutional layers, denoted by the numerical suffix in their names. Therefore, Resnet50 and Resnet101 feature 50 and 101 convolutional layers, respectively [74].

Mask R-CNN involves four key steps: the backbone, region proposal network, classification network, and the mask head. The backbone scans the image using filters to identify potential objects. These objects are then forwarded to the region proposal network for focused attention and proposal of bounding boxes and labels based on object features. Subsequently, the classification network assigns labels to the objects, and the bounding boxes undergo regression for refinement. For segmentation, a mask head identifies the precise shape of the object at the pixel level [76]. Figure 15 provides an overview of this process. The object detection head is where classification occurs alongside additional bounding box refinement, while parallel generating masks through the Mask generation head.

The backbone for Mask R-CNN is the convolutional neural network described at the beginning of this chapter. The region proposal network (RPN) takes an image as input and returns boxes with object proposals [98]. The generation of the proposals is done by placing a set of convolutional layers over the convolutional feature maps near the end of the CNN. Splitting the process into two different tracks, the output of the backbone and the separation into the RPN. The RPN is followed by two fully connected networks for classification and location prediction. These networks are implemented with square convolutional layers and two 1x1 convolutional layers for the classification and location prediction. The RPN in Mask R-CNN is shared with the Fast R-CNN object detection network [76]. The RoI Align procedure discussed in Section 3.4.3 aligns the bounding boxes with the image. The object detection head is taken from Faster R-CNN [50] [75]. The mask branch is presented in Mask R-CNN's paper [76] as a fully convolutional neural network applied to each RoI. Predicting a segmentation mask pixel by pixel.

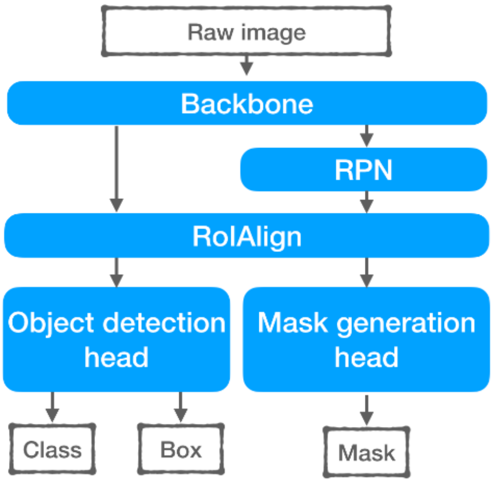


Figure 15 Structure of Mask R-CNN [99]

Cascade Mask R-CNN introduces cascade learning to Mask R-CNN, where the outputs of classifiers undergo multiple progressively refining stages. Each stage builds upon the previous one, aiming for enhanced accuracy and performance [100]. A three-iteration model is depicted in Figure 16, wherein the FPN network operates at the same level as the backbone, extracting features. The bounding box is iteratively passed back to the RoI pooling layer, leading to output generation in C3, S, and B3 after two iterations. The removal of the first two stages would render the model identical to Mask R-CNN.

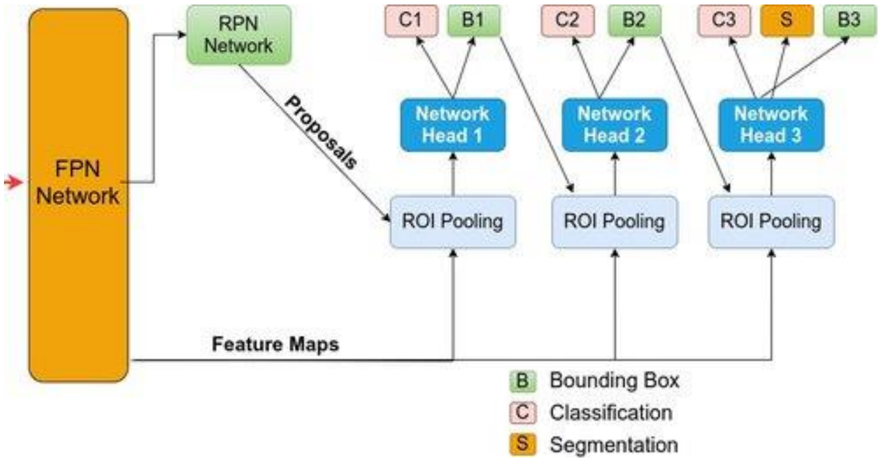


Figure 16 Cascade structure of Cascade Mask R-CNN [101]

ISTR uses a CNN backbone with FPN to construct a feature pyramid [94]. After the feature pyramid are two paths. For one path, RoI Align is used to extract RoI features from the feature pyramid. The other path creates the image's features by averaging and summing the features from the feature pyramid. These image features are position embedded, by cosigning the positions to the values. The image features and the position embedding are inputted into the self-attention module [102]. The self-attention module encapsulates complex relationships among different features. Additionally, a dynamic attention head is included for fusing the RoI and the image features. The combination of the RoI and image features are proceeded to the prediction heads for the predictions. The predictions are made similarly to Mask R-CNN in different heads depending on the task, including a class head, box head, mask head and, for ISTR, a mask decoder. The mask heads output the mask embeddings, which are reconstructed with the mask decoder into the prediction masks. Lastly, a recurrent refinement is done with the predictions by repeatedly updating the prediction boxes. Which refines the predictions and allows for in parallel the processing of the classification and segmentation. Figure 17 presents a simplified overview of ISTR.

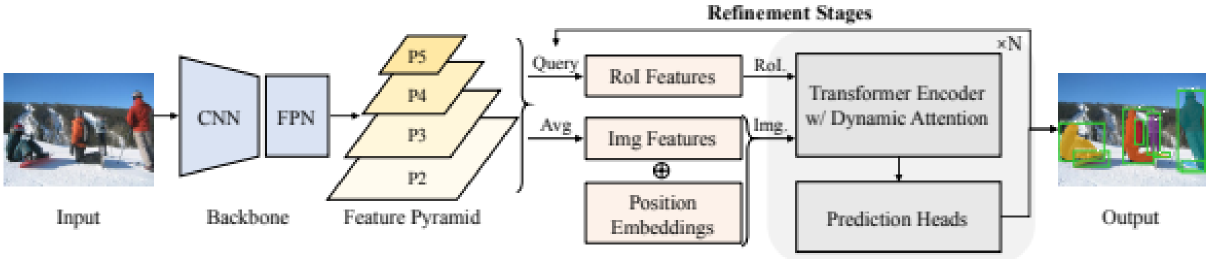


Figure 17 Structure of ISTR [94]

Both Cascade Mask R-CNN and ISTR are utilizing refinement steps, either as cascade learning or recurrent refinement. Compared to Mask R-CNN and Cascade Mask R-CNN, ISTR takes different information from the FPN and uses it to use an alternative approach to providing data to the prediction heads. This alternative approach does allow ISTR to produce masks without relying on bounding boxes unlike Mask R-CNN and Cascade Mask R-CNN.

3.6. Performance evaluation metrics

The assessment of algorithmic performance within computer vision is typically enabled through the utilization of metrics. Originating from the foundational focus on classification tasks in computer vision, the metrics employed are derived from established methods for quantifying the efficacy of classification models. As explained in 3.3, classification models are primarily set up towards determining whether an image depicts a specific object. The outcomes of model decisions are encapsulated in a comprehensive framework known as a confusion matrix [103], as illustrated in Table 6. This matrix systematically captures the correctness of model outputs for each processed image. Subsequently, following the model's testing phase, the confusion matrix serves as a foundational tool for conducting performance assessments tailored to classification models.

Table 6 Confusion matrix

	Predicted positive	**Predicted negative**
Actual positive	True positive	False negative
Actual negative	False positive	True negative

The confusion matrix, as described in Table 6, presents four distinct outcomes based on the model's predictions: True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). A more specific illustration, provided in Table 7, further explains the application of this matrix in the context of images featuring dogs. Here, the top row signifies the model's predictions, distinguishing between a particle and a non- particle, while the first column represents the actual nature of the image, classifying it as either a particle or not.

Table 7 Confusion matrix for images of particles

	Predicted particle	**Predicted not a particle**
Actual particle	TP	FN
Actual not a particle	FP	TN

However, the application of a confusion matrix becomes less straightforward when dealing with instance segmentation, particularly in scenarios such as particle detection. In the realm of instance segmentation, assessments are diverged into object detection and the corresponding segmentation of identified objects. Although not directly employed, the majority of performance metrics for instance segmentation can be translated back to the fundamental principles of a confusion matrix.

Prominent metrics applied to evaluate segmentation performance include mean intersection over union (mIoU), pixel accuracy, average precision (AP), average recall (AR), and the F-score. These metrics collectively contribute to a comprehensive understanding of the effectiveness of instance segmentation models in finding and accurately describing distinct objects within an image.

3.6.1. Mean intersection over union

The intersection over union (IoU) metric quantifies the degree of overlap between a predicted bounding box and the corresponding ground truth bounding box. Mathematically, IoU is calculated as the ratio of the area of overlap to the area of union and is alternatively referred to as the Jaccard index [104]. The resulting IoU value ranges between 0 and 1, where a lower score indicates a poorly selected bounding box with minimal overlap, while a score approaching 1 signifies a bounding box closely aligned with the ground truth. The visual depiction in Figure 18 clarifies the concept of IoU using a predicted bounding box and the associated ground truth.

In the context of evaluating multiple objects' localization, the mean intersection over union (mIoU) is employed. The mIoU considers the IoU for all predicted bounding boxes, offering a comprehensive performance assessment for object localization across the entire dataset.

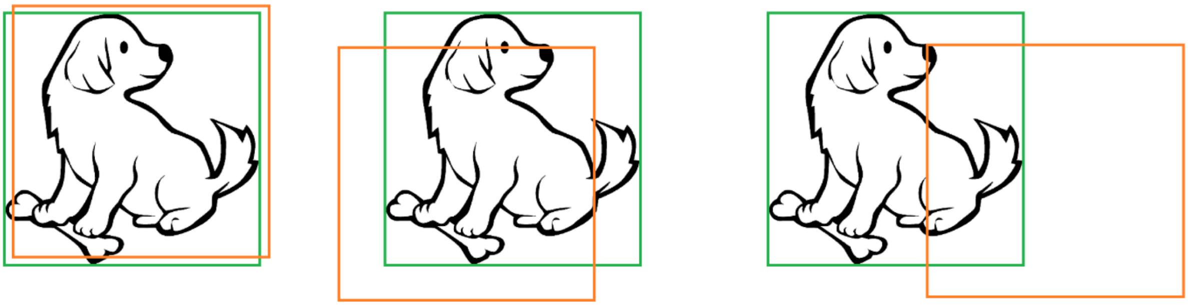


Figure 18 Example of three IoU values for the bounding box of a dog. The IoU from left to right: 0.97, 0.73, 0,29

3.6.2. Pixel accuracy

Pixel accuracy gauges the proportion of correctly labelled pixels relative to the total number of pixels in an image [105]. This metric, however, does not factor in false positives and true negatives. Furthermore, in scenarios involving multiple classes, individual class performances are not taken into consideration. It is crucial to note that pixel accuracy might yield deceptive results if the test image selection is not meticulous. While pixel accuracy predominantly focuses on segmentation, object detection employs a similar metric known as accuracy. In object detection, accuracy assesses the correct identification of objects, dividing this by the total number of objects present in the image. Similar to pixel accuracy, maintaining a balance between classes is imperative for accurate measurements, as false positives and true negatives are not accounted for. To illustrate, consider a scenario where 90% of the mixture comprises component 1 and the remaining 10% is component 2. If the model erroneously labels every particle as component one, the accuracy would still be 90%, even though the practical utility of such a model would be compromised.

3.6.3. Average precision (AP)

Precision, a fundamental metric, is defined as the proportion of correctly identified objects divided by the total number of predictions [106]. As an illustration, suppose a model predicts 10 objects within an image, of which 6 are correct; the precision, in this case, would be 0.6. Precision exclusively provides insights into the likelihood that when the model identifies an object, it is indeed the predicted one. To calculate the average precision (AP), the cumulative sum across multiple images is determined for each object category. In object detection, where models identify and localize various object categories within a single image, the AP is instrumental in evaluating performance across all categories. Consequently, AP is often referred to as mean average precision (mAP), representing the mean of the AP per category and offering a comprehensive evaluation of overall model performance.

The AP metric is frequently integrated with intersection over union (IoU) to assess performance with varying degrees of bounding box overlap. Three IoU thresholds are commonly considered: .50:.05:.95, .50, and .75, with .50:.05:.95 being the primary challenge metric. Another approach involves examining AP concerning the size of objects and categorizing objects based on the area of their bounding boxes. The areas are used to determine size categories, as outlined in Table 8.

Table 8 Various Average Precision measurements

Average precision (AP)	Description
AP	% AP at IoU = .50:.05:.95
$AP^{IoU=.50}$	% AP at IoU = .50
$AP^{IoU=.75}$	% AP at IoU = .75
AP^{small}	% AP for small objects with area smaller than 32^2
AP^{medium}	% AP for medium objects with area between 32^2 and 96^2
AP^{large}	% AP for large objects with area larger than 96^2

Precision can be expressed using the confusion matrix, Table 6: [106]

$$P = \frac{TP}{TP + FP} \tag{2}$$

3.6.4. Average recall (AR)

The recall of a model, often referred to as sensitivity [73], gauges the model's capacity to detect objects. Analogous to average precision, average recall is linked to intersection over union (IoU). Unlike average precision, which is evaluated at various IoU thresholds, average recall focuses solely on the .50:.05:.95 range and is thus omitted in the table description for brevity. Average recall is computed concerning the number of detections and the size of the objects.

Table 9 Various Average Recall measurements

Average recall (AR)	Description
AR^{1}	% AR at 1 detection
AR^{10}	% AR at 10 detections
AR^{100}	% AR at 100 detections
AR^{small}	% AR for small objects with area smaller than 32^2
AR^{medium}	% AR for medium objects with area between 32^2 and 96^2
AR^{large}	% AR for large objects with area larger than 96^2

Recall can be expressed using the confusion matrix, Table 6:

$$R = \frac{TP}{TP + FN} \tag{3}$$

3.6.5. F-score

The F-score, or F1-score, is defined as the harmonic mean of the precision and the recall [107].

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

The harmonic mean, distinct from the arithmetic mean typically used for averages, provides a more nuanced representation of performance assessment [107]. When comparing the two means, the harmonic mean proves to be more indicative in measuring performance.

The arithmetic mean (A) and harmonic mean (H) are defined as follows:

$$A = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\,(x_1 + x_2 + \ldots + x_n) \tag{5}$$

$$H = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}} = \frac{n}{\frac{1}{x_1} + \ldots + \frac{1}{x_n}} \tag{6}$$

For the means of precision (P) and recall (R), the equations become:

$$A = \frac{1}{2}(P + R) \tag{7}$$

$$H = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{8}$$

$$= \frac{2}{\frac{P + R}{PR}} \tag{9}$$

$$= \frac{2PR}{P + R} \tag{10}$$

The harmonic mean can be transformed into the equation seen for the F-score. For example, in particle recognition, with precision at 0.9 and recall at 0.1, the arithmetic mean is 0.5, while the harmonic mean is 0.18, offering a more realistic representation of performance.

The term F1-score is derived from the F measurement [108]. The F measurement uses F_β, where β ranges from 0 to +∞. As β is squared in equation (11), negative numbers do not deviate from their positive counterpart. In the F1-score, β is set to 1:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{11}$$

The F1-score, with β=1, balances precision and recall equally and results in equation (10). Other values of β, such as F_2, $F_{0.5}$ or F_β, allow prioritization of precision or recall based on specific needs. For instance, F_2 places more emphasis on recall, $F_{0.5}$ prioritizes precision, and F_β permits custom balance shifts for specific cases. Values for β between 0 and 1 favour the precision of the model, with values closer to zero providing a stronger relation with the precision. Using β=0 makes the F measurement equal to the precision. On the other hand, moving β further away from 1 and 0, the F measurement shifts towards the recall. Assigning a huge value to β makes the equation equal to the recall.

The F1-score is used in this study as there is an equal importance on both the precision and recall. The metric considers both missing and misclassified identifications, influencing the performance evaluation. Precision and recall contribute equally to the assessment, ensuring a balanced evaluation of models based on β=1. The F-score utilizes average precision for precision and average recall for recall, both measured with IoU = .50:.05:.95 for bounding box overlap.

3.7. Conclusion

In the field of machine learning, manual crafting of features poses a challenge, particularly when dealing with highly similar objects, constraining the model to adhere closely to learned images. Conversely, deep learning eschews explicit feature engineering, relying on a substantial number of examples. This abundance of examples enables deep learning models to generalize patterns from observed images, enhancing predictive capabilities when confronted with conditions that diverge from the learned scenarios.

The process of particle recognition necessitates a mechanism for the computer to interpret complex mixtures. In the context of presenting the environment to a computer through cameras or sensors, the domain of computer vision comes into play. Convolutional Neural Networks (CNN) are prevalent in computer vision, serving as widely employed networks for decision-making based on images. The specific task at hand is segmentation, a process that assigns labels to pixels corresponding to individual objects. The segmentation of the entire image yields distinct groups of pixels representing various objects, which can be further transformed into coloured masks. The training methodology employed for this segmentation task is supervised learning, requiring manually labelled images to impart the knowledge necessary for recognition and replication. Throughout the training phase, the model utilizes the Gradient Descent algorithm to iteratively refine its behaviour.

Three models are selected for a performance comparison on our dataset composed of granular materials. The models have to comply with the requirements of the tasks instance segmentation and to be publicly available. To select the three models, there is looked at the potential and relevance of the models. The potential describes the hypothetical performance a model could achieve. The relevance is seen as the popularity and reoccurrence of the model over time. Resulting in three models to be selected. The models Mask R-CNN, Cascade Mask R-CNN and ISTR are taken for further testing.

The evaluation is done with metrics to describe the performance. There are several options when considering metrics, the most commonly presented metrics are the average precision (AP) and average recall (AR). To obtain a single comprehensive value to present the performance, metrics such as the F-score are used. As the F-score comprise of the common metrics AP and AR, the F-score is considered during the evaluation of the selected models.

4. Application of artificial intelligence for quantifying segregation

The models chosen in section 3.4 will be applied to a mixture of granular materials. The purpose of the models is to find all particles, segmentate the particles and identify the correct material. Finding the best suited model for the task is done by training all three models and running performance tests. After which the selected model will be applied to extract data from the granular mixture to estimate the segregation.

4.1. Model selection for segregation measurements

In the pursuit of finding a suitable model for the task of assessing segregation in granular materials, an exhaustive examination of various models was undertaken. The exploration initiated with an analysis of machine learning models, focusing specifically on the subset known as deep learning, as elaborated in Section 3.1. This decision led to that deep learning, owing to its inherent capacity for pattern recognition, would be a justifiable choice, particularly given the inherent similarity of the objects within this study requiring identification. Subsequently, in Section 3.2, a detailed exploration of deep learning within the domain of computer vision was conducted. This exploration shows the intricacies of instance segmentation tasks and the requisite learning setup, necessitating a dataset with annotated examples for segmentation purposes. The significance of a substantial dataset, characterized by numerous examples and diverse data variations, was underscored, recognizing the enhanced performance achievable under a broader spectrum of conditions. The training procedure adhered to Gradient Descent principles, as expounded in Section 3.2, necessitating an iterative approach to converge towards an optimal solution. To ensure equitable comparison, a uniform constraint of 40 thousand iterations (or 222 epochs) was imposed for maximum training, fostering consistent conditions among the models. Post-training, the models underwent evaluation using an unseen image to gauge their performance, and the ensuing assessment was quantified through evaluation metrics, as detailed in Section 3.6. These performance metrics, serving as discernible indicators of expected model proficiency, facilitated a comprehensive observation of model behaviour when confronted with an image. In the event of noticeable performance gaps, the prospect of considering an alternative model was contemplated.

4.1.1. Dataset creation

A dataset, a fundamental component for training models, is a curated collection of data integral to the supervised learning method in computer vision. Specifically, for deep learning models in computer vision, this entails a set of images accompanied by an annotation file explaining the object attributes and their spatial configurations.

The image acquisition process involves the deliberate formation of heaps. For the creation of the heaps, the materials are loaded into a cylinder. The cylinder is lifted to allow to materials within to spread out from underneath. Resulting in the forming of a heap. The heap was thereafter captured from a top perspective to create images such as Figure 19. The process is conducted upon an initially white cloth. This methodology facilitates the expeditious collection of particles, yielding a diverse set of images from various heaps. These images are used for multiple purposes: they constitute the training data for the model, contribute to performance assessments, and function as test cases. Importantly, to prevent any potential model bias, distinct sets of images are allocated for training, performance evaluation, and testing, ensuring a lack of overlap between these tasks. The distinct sets of images are manually separated. The method of using separate sets for training and performance evaluation is referred to as cross-validation.

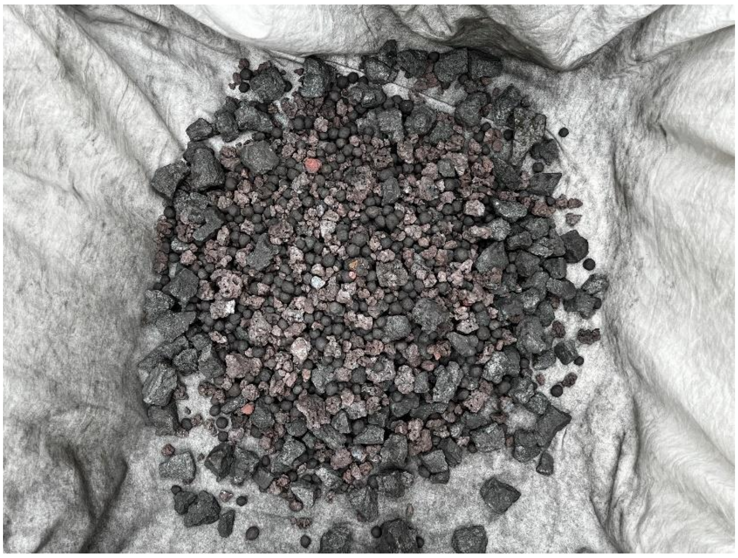


Figure 19 Image taken of the created heap

For model preparation, a training dataset comprising 13 thousand particles was employed, consisting of 6.5 thousand sinter, 2 thousand coke, and 4.5 thousand pellet particles. The dataset exclusively holds images of particles in a mixture, providing a realistic environmental context for the recognition task. In adherence to the principles of deep learning, which thrives on extensive datasets, the substantial quantity of objects enhances the model's ability to generalize and predict unforeseen variations in object characteristics.

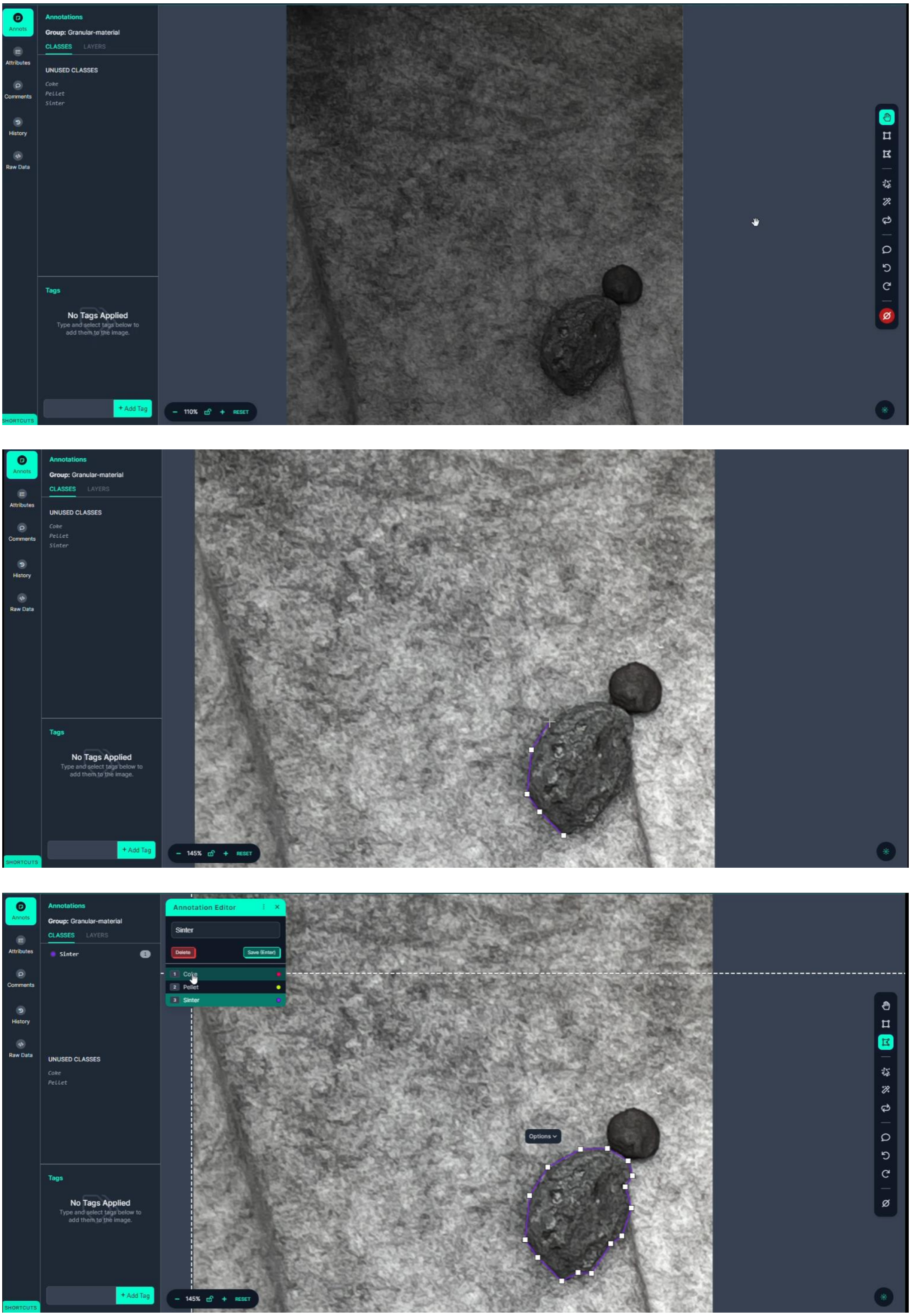
The accompanying annotation file constitutes a crucial aspect of dataset preparation, containing a comprehensive list of objects per image. Each object is characterized by its material type and a set of coordinates. Manual annotation, accomplished by enumerating a list of coordinates and connecting them sequentially to outline the object shape, is the traditional approach. However, alternative tools, categorized as commercial or open source, offer more intuitive annotation methodologies. Open source tools, while fully capable of annotations, may lack the convenience provided by commercial tools. An illustrative example is the incorporation of selection assistance in commercial tools, which, particularly in segmentation tasks, proposes object contours based on edge detection or object detection AI. While these suggestions may necessitate minor adjustments for precise inscriptions, they notably expedite the annotation process, particularly for irregular shapes. The end-user, functioning as a quality control entity, validates the suggested annotations, thereby ensuring dataset quality. In the absence of such assistance, annotations involve placing dots around the object, connected by straight lines to enclose the object. For irregular shapes, this manual dot-placement process can be considerably time-consuming.

As the number of annotation tools available is large, Table 10 presents a handful of examples, including both open-source and commercial options. All listed tools featured facilitate segmentation annotations, as there are tools specialized for image classification and/or object detection. The selection of a specific tool depends on individual preference. In this study, Roboflow, an effective tool for dataset preparation in the context of granular material segregation is used.

Table 10 Annotation tools

Name	Open-source
CVAT [109]	Yes
Label-studio [110]	Yes
VIA [111]	Yes
Labelbox [112]	No
V7labs [113]	No
Roboflow [114]	No

The process of annotating is described in the following images. First load in the image to be labelled. The classes that are going to be appointed to the objects could be done beforehand or when assigning an object. To mark the mask, a polygon is drawn around the object by placing vertices on the edge of the object. The polygon is used as a reference for the learning process of the model. Loose annotations would therefore result in the outputs also loosely encompassing the object. After completing the polygon, a class is assigned. In the case of the example, the classes are already made and the corresponding class could be selected. The steps of creating a polygon and assigning a class are repeated till all particles are marked. A visual representation of these steps is shown in Figure 20 and ends with an overview of the reference masks.



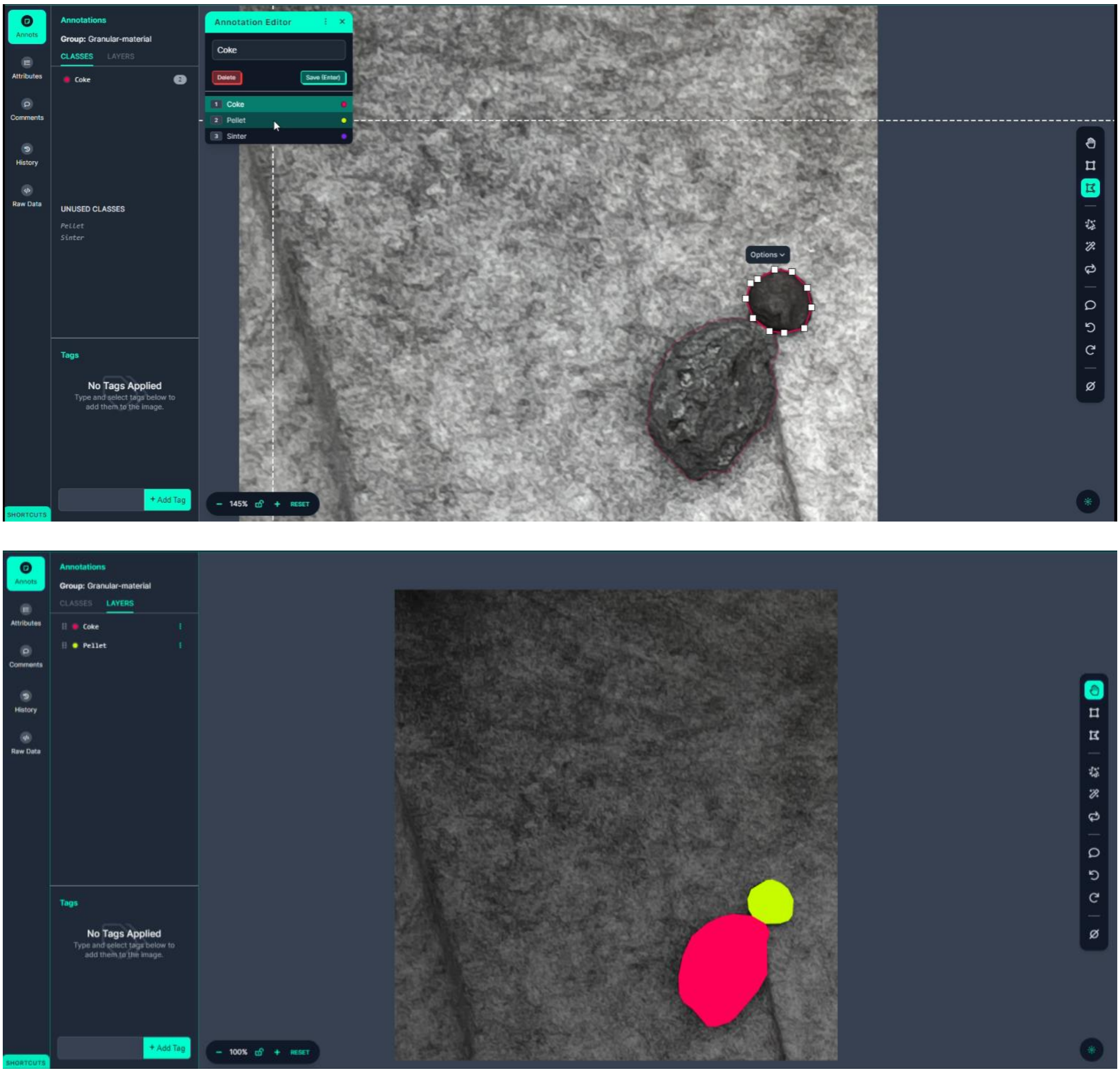


Figure 20 Steps of annotating in images in Roboflow [114]. In order: Loading in an image, create a polygon around the object, assign a class to the object, repeat for all objects and the reference masks for this annotated image.

4.1.2. Training setup

The training of the models is done in python on the aforementioned dataset. To control the learning process, parameters in the form of hyperparameters are used. Hyperparameters are used to describe any process, component and step in the learning process and the model. Therefore, a small number of hyperparameters are given in Table 11, while others are in Appendix B. The first four hyperparameters describe the learning iterations. MAX_ITER prescribes the maximum of iterations the model does on the images. BASE_LR is used to indicate the size of change the model is permitted to apply based on the back propagation. The following two hyperparameters introduce a stepdown in the learning rate with GAMMA the magnitude of the step and STEPS the iteration when the steps occur. Lastly, the EVAL_PERIOD is the interval for the evaluation on the evaluation dataset to track the progress of the learning. Directly noticeable is the different BASE_LR for ISTR. This is set lower to prevent the divergence of the solver.

Table 11 Training hyperparameters of the three models

Setting	Mask R-CNN	Cascade Mask R-CNN	ISTR
MAX_ITER	40 000	40 000	40 000
BASE_LR	0.001	0.001	5e-5
GAMMA	0.1	0.1	0.1
STEPS	(36500, 38500)	(36500, 38500)	(36500, 38500)
EVAL_PERIOD	500	500	500

Besides setting up the hyperparameters for the models, the images need to be adjusted. Given the constraints of deep learning models regarding maximum and minimum input sizes, preprocessing of inserted images is necessary to ensure compliance with these limits. Two primary methods, downsizing and dividing the image into smaller sections, are considered. Downsizing enables the acquisition of information in a single output that covers the entire image, while splitting the image retains details captured by high-resolution cameras, distributed across several sub-images. Prioritizing the preservation of details, the decision is made to split high-resolution images into smaller sections, as depicted in Figure 21. To determine the size of these sub-images, alignment with the size of images in the dataset is recommended, given that the model anticipates such conditions. The mean size of the dataset images is 600 by 550, while the size used by the ResNet backbone is 224x224 [74]. Consequently, opting for a much larger size would necessitate downsizing, risking partial framing of objects or loss of context. However, in the context of a particle mixture, this limitation is mitigated as the particles are small, and contextual information is derived from surrounding particles. Considering the additional constraint of dividing an image into sections, the size of the sections is set to 504 x 504. For the captured images, this results in 48 sections of 6 by 8, which are subsequently employed to compare particle concentrations for segregation analysis.

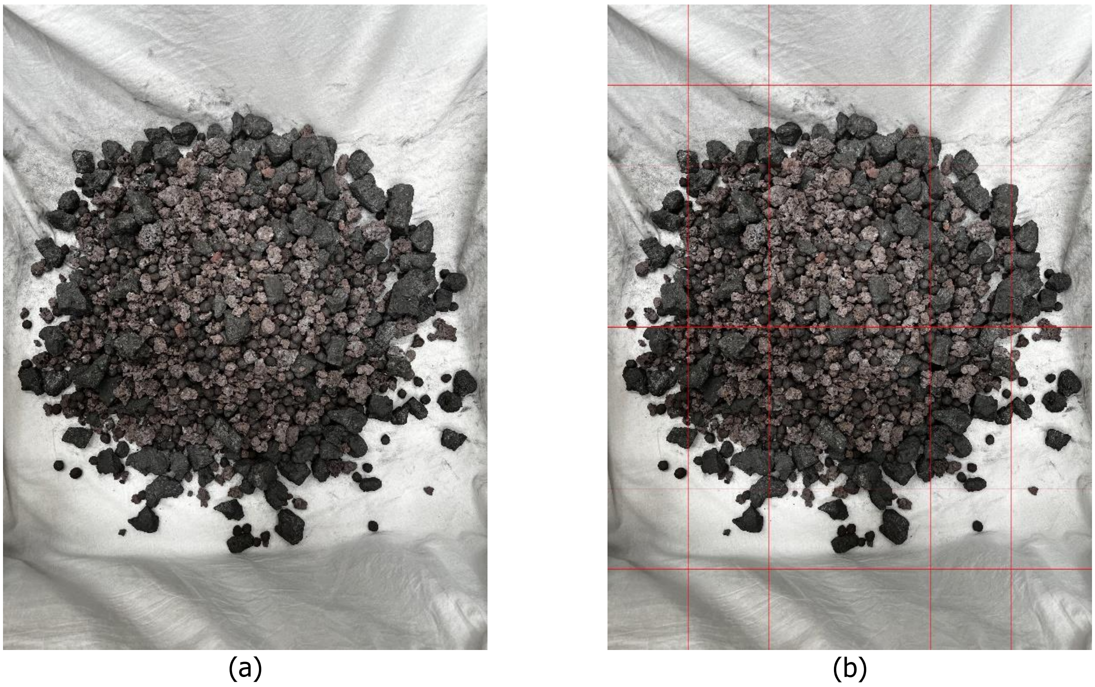


Figure 21 Unused image in the training process utilized for the visual analysis, (a) image of the heap, (b) the division of the sections

The input image undergoes processing through the models, generating a significant amount of information. Since the models are specifically designed for instance segmentation, section 3.3, the expected outputs include:

- Identification of particle types within the image.
- Bounding boxes denoting the locations of identified particles.
- Masks portraying the pixels correlated to individual particles.

For the evaluations, most of the information was directly provided. Between training cycles performance assessments are done to show the progress of the training. The numbers created during the last assessment which occurred after the final iteration are used for the numerical performance evaluation in the following section. The visual analysis takes the outputs responsible for the identification of the particle and the masks. By combining the two outputs, the masks are coloured per material. The materials coke, pellet, and sinter are represented for all coming images by blue, red, and green, respectively. The colours make the predictions from the models easily interpretable. Missing particles stand out by the lack of colour, while wrongly identified particles are found by comparing the predictions with the original image.

The following part explores the capabilities of three models to analyse a mixture composed of coke, sinter, and pellet materials. These materials exhibit common visual features, including shared colours and shapes, with the distinctions becoming more nuanced under conditions with over or underlighting. This inherent complexity makes the mixture a valuable subject for experimentation with a deep learning model designed for recognition.

4.1.3. Numerical performance evaluation

The primary metric used for evaluating performance is the F-score, which considers both precision and recall. The balance between precision and recall, reflected in the F-score, is set to 1 due to the similar importance of these two metrics, as discussed in Section 3.5.5. Additionally, Average Precision (AP) and Average Recall (AR) are provided as they are commonly employed metrics in assessing deep learning model performance. Table 12 displays the performance results of Mask R-CNN and Cascade Mask R-CNN, with minimal differences between them. Notably, ISTR performs noticeably worse under the specified conditions, suggesting that Mask R-CNN is the most suitable for the given task among the three models.

Table 12 Scores of the trained deep learning models

Model	F1 score	AP	AR
Mask R-CNN	0.69	0.67	0.70
Cascade Mask R-CNN	0.66	0.63	0.69
ISTR	0.25	0.21	0.29

4.1.4. Visual performance analysis

For the visual examination, an unused image, shown in Figure 21, created during the dataset creation process but not included in it, is provided as input to the models. This image is processed by the models with the task of identifying particles and marking the corresponding pixels. The output is an image with marked pixels associated with different materials, and the particles are colour-coded (blue for coke, green for sinter, and red for pellet), as shown in Figure 22.

The performance of the models aligns with the predictions from the numerical performance evaluation. Mask R-CNN and Cascade Mask R-CNN show somewhat similar performance, while ISTR significantly deviates. This is evident from the particles not marked with the expected colours (red, blue, or green). A noteworthy distinction between Mask R-CNN and Cascade Mask R-CNN is observed, particularly concerning larger sinter particles. Mask R-CNN appears to face challenges in identifying these particles. When considering predictions with the confidence threshold lowered as shown in Figure 23, Mask R-CNN marks all particles. The precision and the recall of the output of the model are tied to the confidence threshold [115]. The model has uncertainties about the correctness of all predictions which is referred to as the confidence score given as a percentage or a number between zero and one. Lowering the confidence threshold allows predictions with lower confidence to be outputted. The predictions with lower confidence are more likely to be incorrect. For the recall, the threshold results in the elimination of the uncertain predictions, thereby reducing the number of particles that are detected as only particles with a high probability are passed along. The balance is to be calibrated towards the intended use case [116]. Figure 22 shows the predictions with a confidence threshold of 0.8 while Figure 23 is lowered to 0.4.

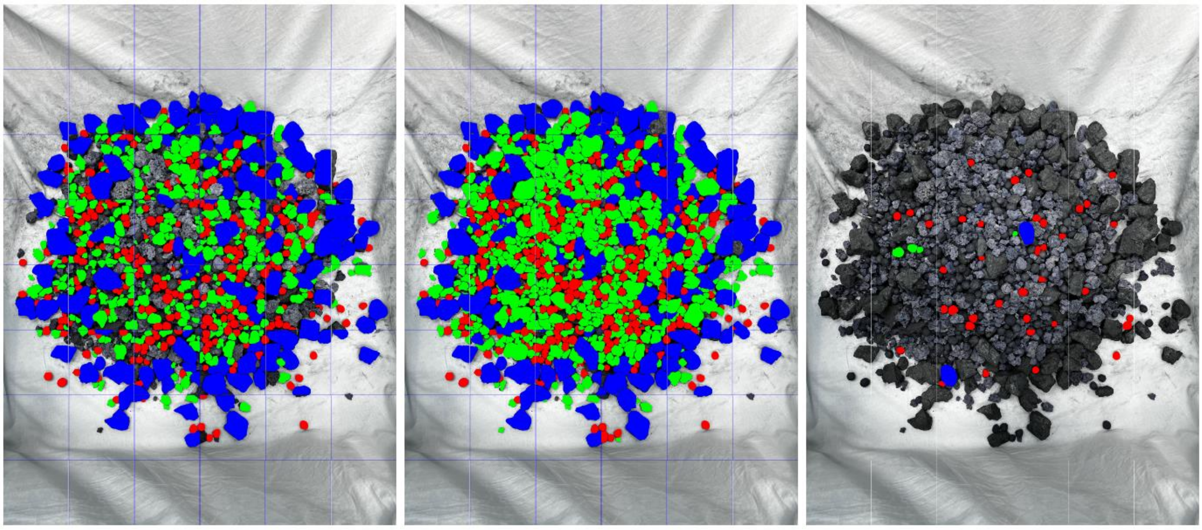


Figure 22 Predictions from the models, from left to right Mask R-CNN, Cascade Mask R-CNN and ISTR

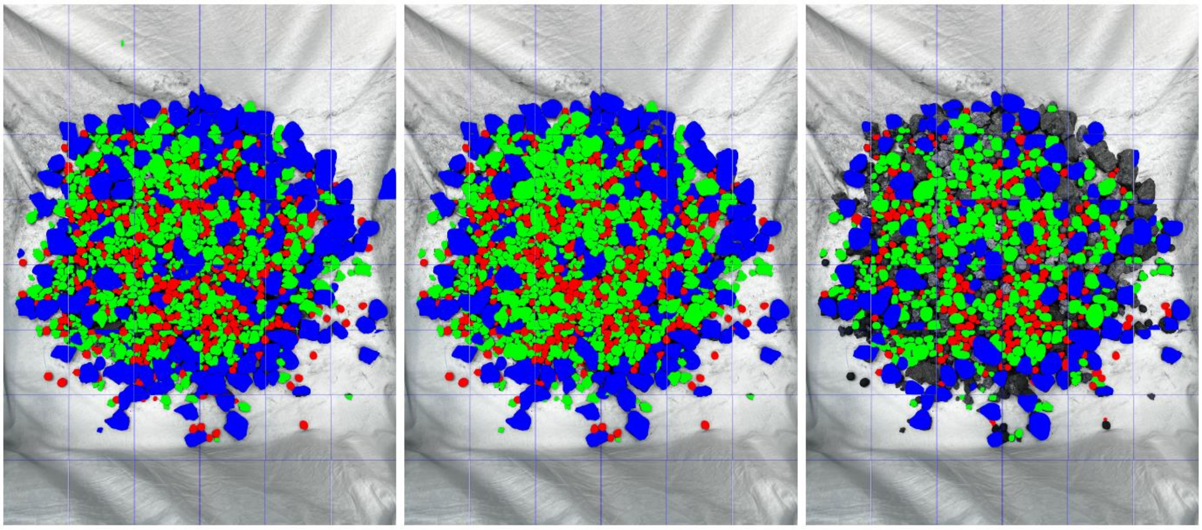


Figure 23 Predictions from the models with confidence threshold lower, from left to right Mask R-CNN, Cascade Mask R-CNN and ISTR

4.1.5. Conclusion

A noteworthy observation is the inconsistency between the numerical and visual assessments. While the numerical evaluation allows for the ranking of models based on performance metrics, the visual examination reveals subtle aspects beyond numerical scores. For instance, the challenges encountered with larger sinter particles using Mask R-CNN. The inconsistency between these two analyses underscores the significance of not overly relying on a singular evaluation method.

In light of both analyses, the selection of ISTR is excluded as it significantly underperforms compared to the other models. The performance between Mask R-CNN and Cascade Mask R-CNN appears comparable in numerical evaluation, with a marginal 0.03 difference, predominantly on the AP side of the equation. However, the visual analysis exposes Mask R-CNN's struggles with larger sinter particles, indicating a substantial issue in identifying this class. Consequently, within the defined constraints, Cascade Mask R-CNN emerges as the most effective model.

The separation between numerical analysis and the visual representation of Mask R-CNN can have several causes. Firstly, the confidence for the numerical assessment is not included. A significant drop in performance can be seen solely based on the confidence score threshold [117]. The size and variation in the evaluation dataset are an additional potential cause. When the size and variation are not sufficient, the evaluation cannot assign representable values for all cases. Therefore, the images used for the evaluation dataset might have been more favourable for Mask R-CNN compared to Figure 21, providing a higher score than appropriate. A similar argument could be given for Cascade Mask R-CNN or ISTR, the used images might have been (un)favourable.

The underperformance of ISTR was expected due to the constraints. ISTR used a 5e-5 as it tended to diverge at larger values, compared to Cascade Mask R-CNN and Mask R-CNN which used a 1e-3, which is twenty times bigger. Therefore, the adjustments to the weights are smaller and likely require more iteration to reach a similar performance. Besides the learning rate, transformers work better with larger datasets as transforms lack inductive biases compared to CNNs [118]. Transformers do not have any prior knowledge of dealing with visual data, therefore requiring larger training data to get started [93]. As a result, transformers need significantly longer training times. Even though transformers have the potential to be the best models, the conditions under which the selection was held were not in their favour.

4.2. Analysis of extracted measurements for segregation

The following sections explore the capabilities of the selected model to analyse the mixture composed of coke, sinter, and pellet materials. The emphasis is on the application of measuring material and size segregation based on the model's predictions. For the application of the model, Figure 24 presents a new case.

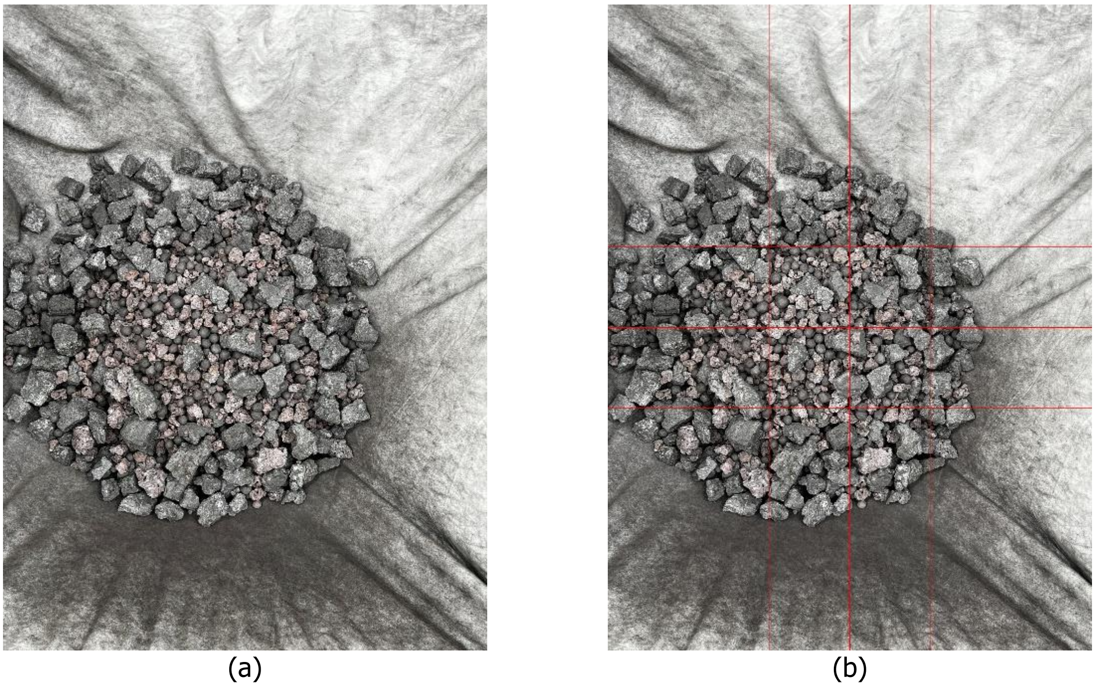


Figure 24 Unused image in the training process used for extraction of segregation measurements (a) image of the heap, (b) the division of the sections

Leveraging the output from the model opens up various possibilities. Initially, a clear overview of the observations of the model is created by grouping individual materials. For the Figure 24, the grouping results in Figure 25. Similar to in section 4.1.4 the materials coke, pellet, and sinter are highlighted by the colours blue, red, and green, respectively. Beyond visual clarity, calculations can be performed with the collected information, offering the possibility to measure the segregation of components within the image. Subsequent sections will delve further into the potential applications of measuring material and size segregation from a single image within the mixture. Consequently, all presented images, numerical data, and graphs correspond to the same experimental scenario.

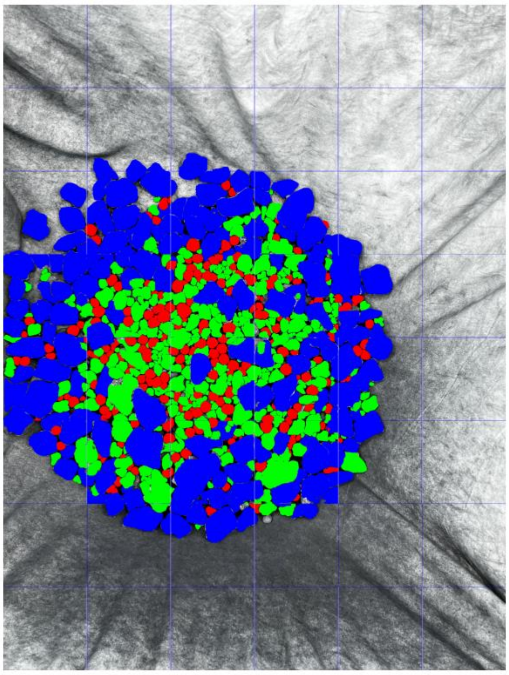


Figure 25 Particle masks coloured based on material, coke - blue, pellet - red, sinter - green

In addition to the colour-coded image representation, the recognition process can be extended to encompass a monochromatic version, encompassing all masks as well as masks exclusively dedicated to each of the individual materials, as presented in Figure 26.

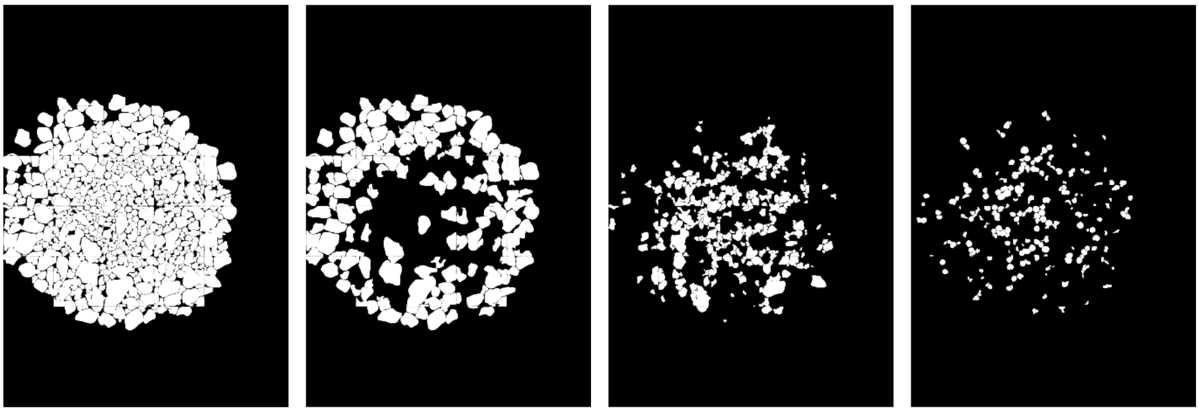


Figure 26 Particle masks. From left to right, all materials, coke, sinter, pellet

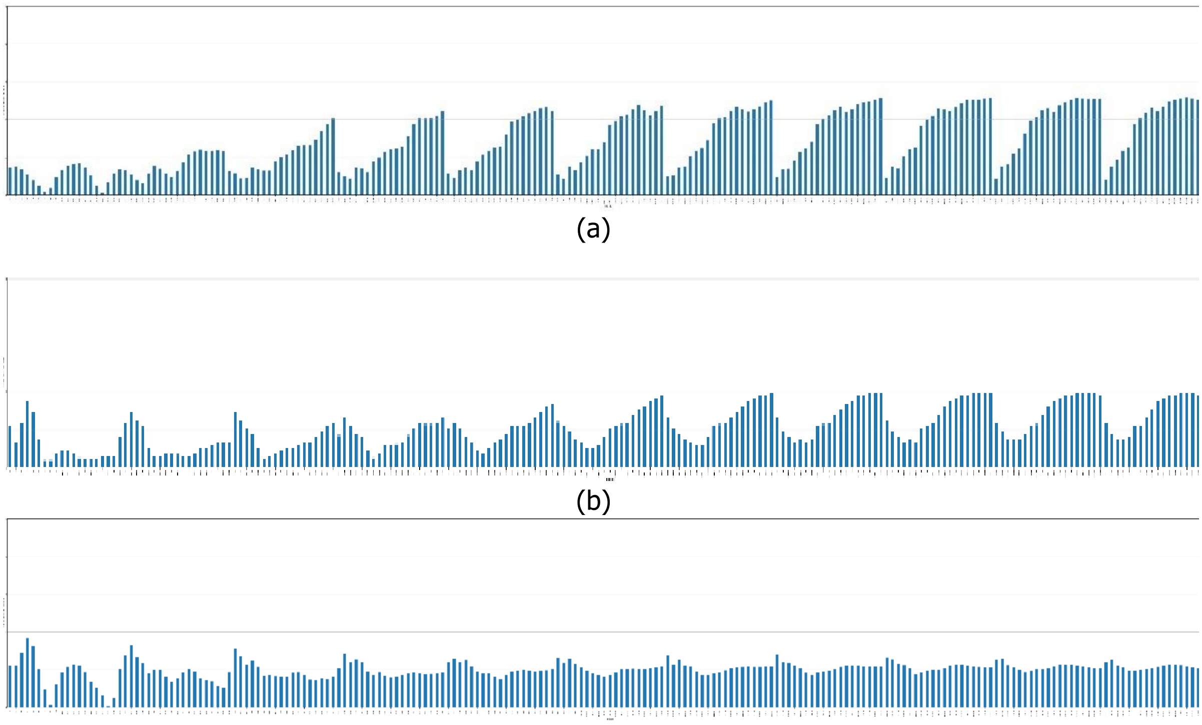
4.2.1. Results: material segregation

From the information provided by the model, it is possible to extract details about the type, location, and size of particles, employing a similar approach used for generating images displaying masks. Given that segregation is a relative measure based on overall presence in the mixture, the image is partitioned into smaller sections to calculate a segregation index (SI). This division can be based on the splits from the input image or defined by specifying the pixels per section. Within each section, the pixels corresponding to masks for each material are tallied. By dividing the pixel counts over the pixels in the section and comparing them within different sections, the segregation in the image can be assessed using equation (1). This formula produces a score between 0 and 0.5, where 0.5 indicates complete segregation and 0 represents perfect mixing.

Table 13 Segregation index for material segregation

Material	Segregation index
Coke	0.22
Sinter	0.15
Pellet	0.07

Another approach is to manually define the sections in the image instead of relying on the division for the processing of the image. Manually defining the section enables one to look at the materials in a more relevant manner. As for the heap, an interesting feature would be the outwash of the materials. To measure the segregation due to the outwash, the sections need to have a circular pattern. A centre point is placed on the middle of the heap and five radii are used to create the sections. Resulting in multiple rings of materials. By utilizing the separate masks from Figure 26, each pixel is given a value representing the material. Concentrations of the materials are determined based on the values of the pixels. The concentrations are used in equation (1) to obtain the segregation indices for the material segregation, as shown in Table 15. The process gives several parameters to tune to ensure the accuracy of the measurements. Besides the centre point which is set to the middle of the heap, there are the number of rings and the radius of the rings. By exploring the measurements at different values for the rings and radii, low numbers of rings together with small radii provide inconsistent measurements as seen in Figure 27. Figure 27 (a) gives the SI for coke where with each configuration the radius increases by ten pixels. After twenty increases of the radius, a ring is added and the radius is set back to the beginning. This is done for eleven additional rings, starting from one ring and ending with twelve rings. From ring four a clear shape of the graph is being formed. A similar response is observed for sinter in Figure 27 (b) while pellet becomes fairly uniform for all configurations.


(a)

(b)

(c)

Figure 27 SI for (a) coke, (b) sinter, (c) pellet for multiple radii and number of rings configurations. Every iteration the radii increase by 10 pixels, after 20 iterations the radii reset and a ring is added to the configuration.

Hence the graphs became more stable for the larger number of rings, a more detailed look was taken at the total covered area of the rings. The difference between the inner and outer radii of the ring is kept small, at fifty pixels. The number of rings increased from one ring to thirty rings. A similar shape in the SI graph is observed, Figure 28 shows the same trends as seen in Figure 27. Laying the graph, Figure 28, next to an image of the rings, Figure 29 (b), shows that the dip in coke's SI curve around configuration 23 correlates with the edge of the materials. The rings after the edge do not add any particles but influence the measurements by providing sections with low concentrations of material. This affects both the mean and adds a large number for the difference of the mean, resulting in a larger SI. This effect is seen in Figure 28, as coke and sinter start to rise significantly. The early rise of the sinter SI is due to the low presence of the sinter in the outer parts of the heap as seen in Figure 29 (a).

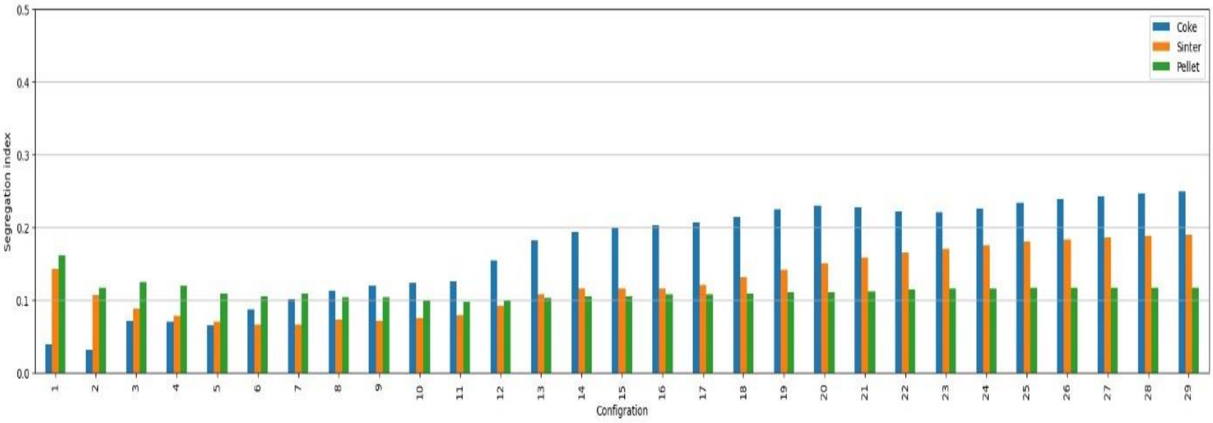
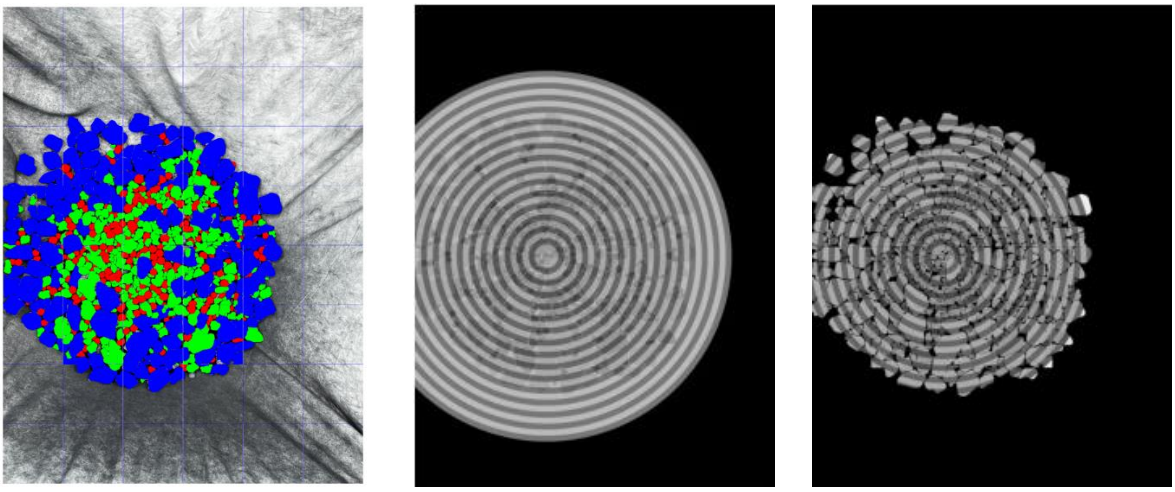


Figure 28 SI for coke, sinter and pellet. Configuration reflects the number of rings with radius increases of fifty pixels



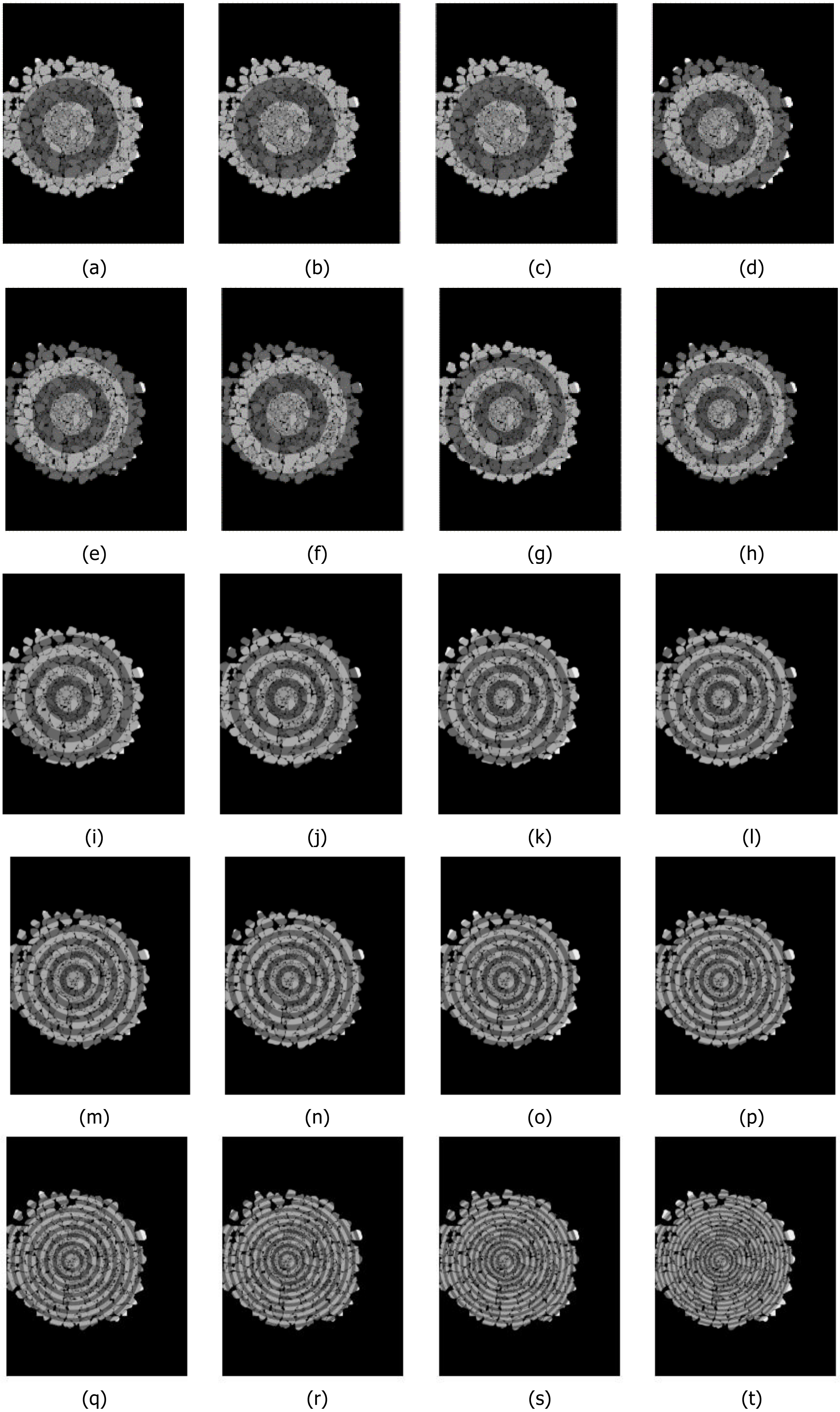
(a) (b) (c)

Figure 29 (a) particle masks colourized, Coke - blue, pellet - red, sinter – green (b) 29 rings over the heap (c) configuration 22, clear white parts of the particles are not included in the measurements

For the number of rings and the radius, the main concern is the number of pixels with cloth included in the calculations. The outer rings go from mostly particles to entirely cloth. The relevance of the most outwards particles needs to be decided. In other words, an outer parameter, the distance from the centre that is taken into consideration while the rest is ignored, for the measurements is to be set. In this thesis, the transition from materials to cloth is taken into the measurements. Therefore, the outer parameter will be on the edge of the heap, which corresponds to configuration 22. Configuration 23 covers the edge of four particles, including barely any particles. Figure 29 (c) shows the particles included in the measurement in grey scales while the unused particles are white. Therefore, the outer radius of the total area is set to be between 1200 and 1300 pixels. With the outer radius of the total area, several combinations of rings and radii are created and shown in Table 14. The areas of all configurations are shown in Figure 30, providing an impression of the sections for the segregation measurements.

Table 14 Configuration of rings and radius differences for a total area radius between 1200 and 1300 pixels

Configuration	**Rings**	**Radius differences**	**Total area radius**
1	2	390	1220
2	2	400	1250
3	2	410	1280
4	3	290	1210
5	3	300	1250
6	3	310	1290
7	4	240	1250
8	5	200	1250
9	6	170	1240
10	7	150	1250
11	8	130	1220
12	9	120	1250
13	10	110	1260
14	11	100	1250
15	12	90	1220
16	14	80	1250
17	16	70	1240
18	19	60	1250
19	23	50	1250
20	28	40	1210
21	29	40	1250
22	30	40	1290
23	38	30	1220
24	39	30	1250



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

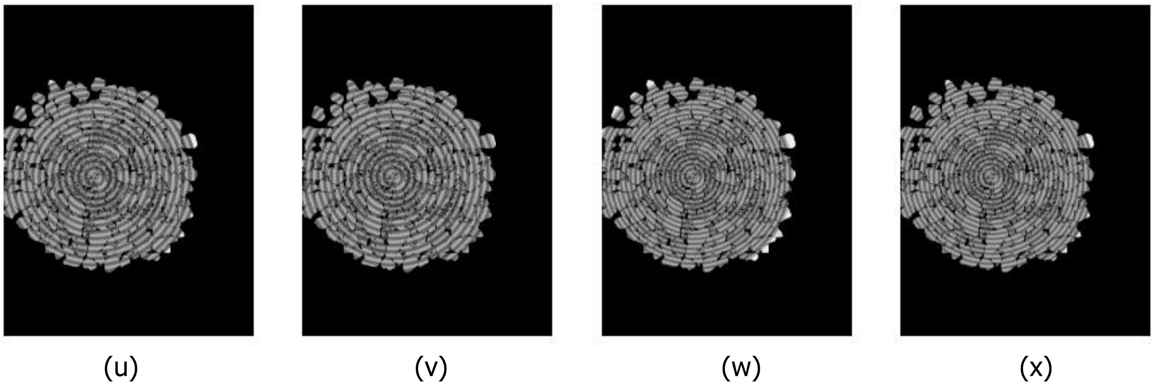


Figure 30 Area of the particles covered visualized for all configurations,
(a) 1 (b) 2 (c) 3 (d) 4 (e) 5 (f) 6 (g) 7 (h) 8 (i) 9 (j) 10 (k) 11 (l) 12 (m) 13 (n) 14 (o) 15 (p) 16 (q) 17 (r) 18 (s) 19 (t) 20 (u) 21 (v) 22 (w) 23 (x) 24

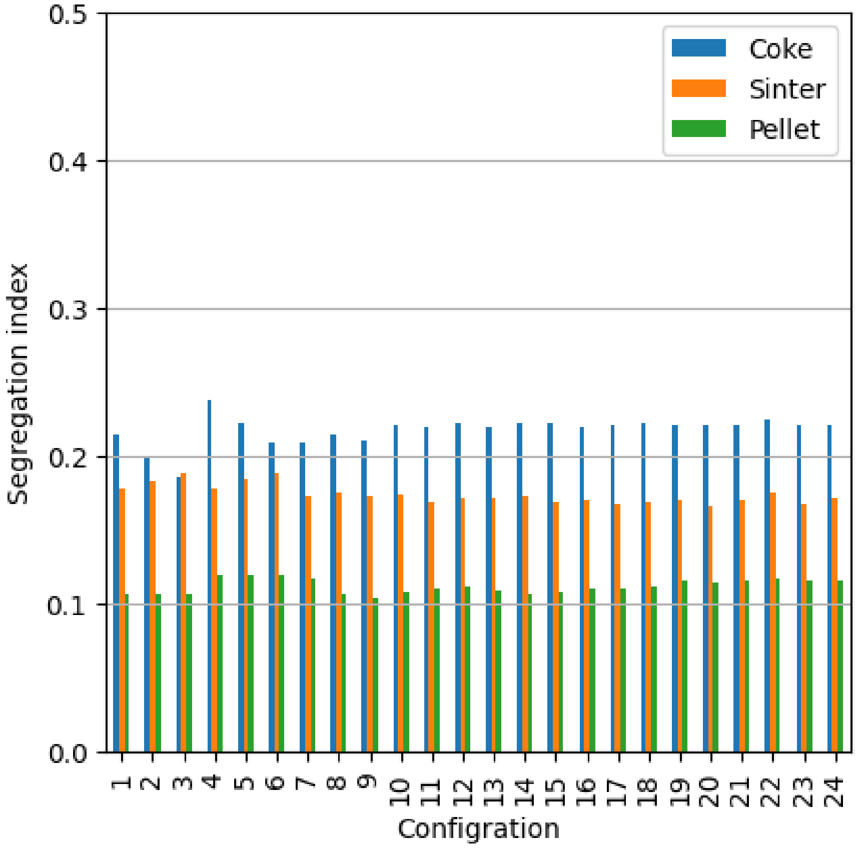
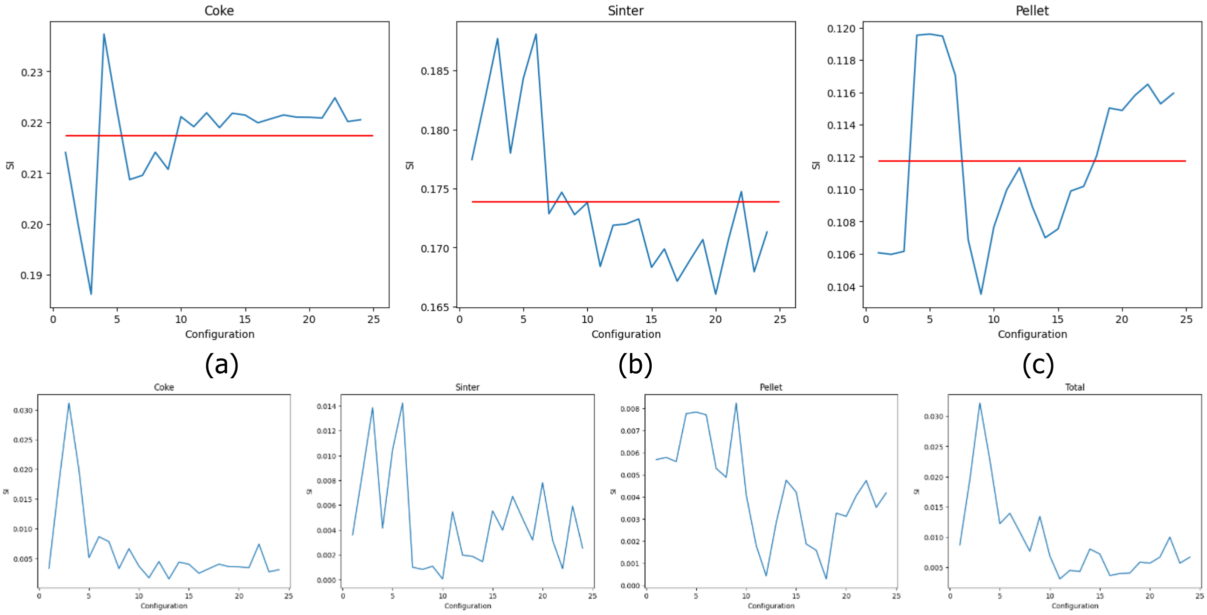


Figure 31 SI obtained from the configurations for Table 14

The segregation measurements of the configurations are shown in Figure 31. As the area of the measurements is consistent, the measurements tend to be fairly close to each other. The configurations with the smaller number of rings deviate from the consistency seen in the later configurations. Plotting the segregation measurements against the mean of the measurements shows a relatively large offset for the configurations 2 to 9. While configuration 1 is close to the mean, the lesser number of rings makes it highly sensitive to minor changes in the measurements. This can be seen in the increase of the size of the rings, configuration 2 and 3 where the results drastically change. The offsets per material are shown in Figure 32 combined with a combined offset over the three materials. The graph of the combined offset in Figure 32 (g) settles below a deviation of 0.01 at measurement 10.

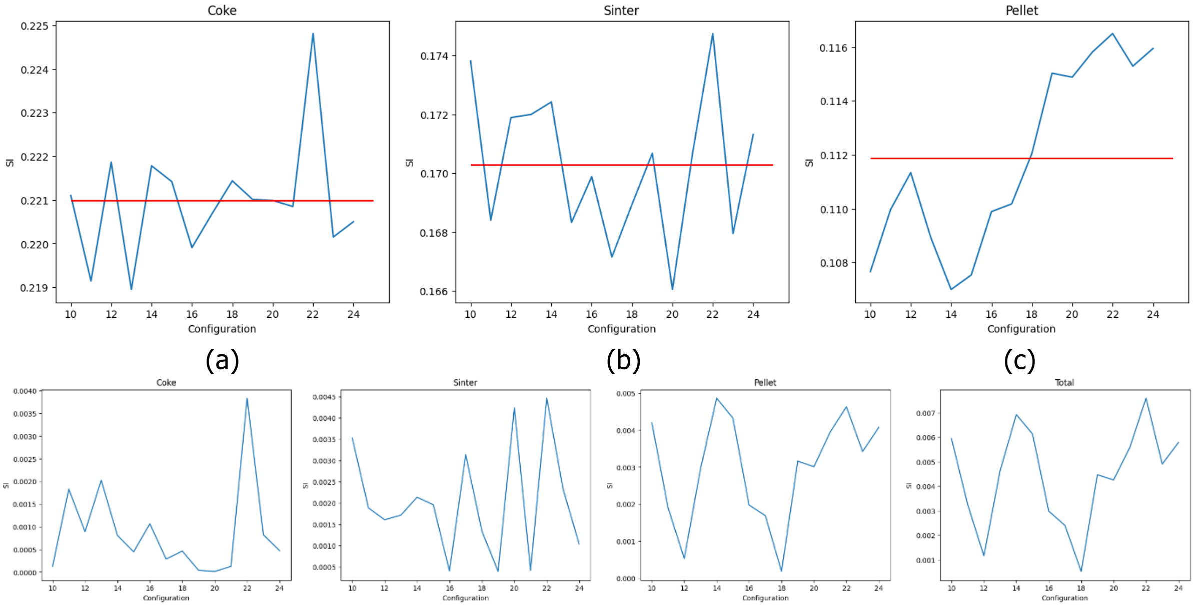


(a) (b) (c)

(d) (e) (f) (g)

Figure 32 Comparison of the measurements from the configurations. (a), (b) and (c) compare the measurements for Coke, Sinter and Pellet with in red the mean of the respective measurement. (d), (e) and (f) show the deviation from the mean for each measurement with (g) being the combination of all three for each configuration.

As configurations 1 to 9 are highly sensitive or have large offsets, configurations 1 to 9 are excluded from the comparison. The exclusion is to reduce the effect configurations 1 to 9 have on the mean of the measurements. The exclusion of configurations 1 to 9 does not influence the result as seen in Figure 33 besides zooming in. Configuration 18 provides the lowest error to the mean. While the optimization of the configuration may make this measurement better, it is image or case specific. The most important aspect is the determination of the area to be measured. In the case of the presented heap, a method is utilizing circles. However, with images fully filled with particles, other shapes are to be considered. Additionally, the location of the centre of the radial measurements influences the measurements when approaching the edge of the materials. As offsets translate to unnecessarily taking empty pixels into consideration for the segregation. As seen in Figure 27, there should be enough rings to take representative segregation measurements. The largest difference from the mean in Figure 33 has the size of 0.005, which is a hundredth of the maximum value of 0.5 for a completely segregated mixture.



(a) (b) (c)

(d) (e) (f) (g)

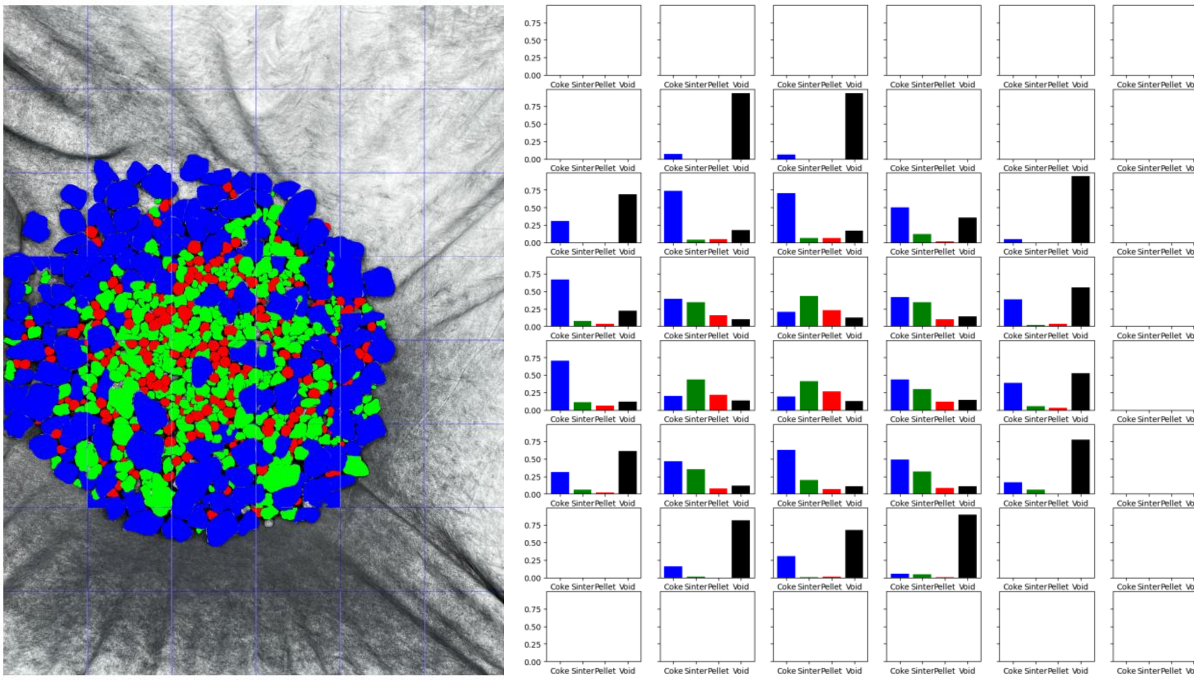
Figure 33 Comparison of the measurements from the configurations with the exclusion of configuration 1. (a), (b) and (c) compare the measurements for Coke, Sinter and Pellet with in red the mean of the respective measurement. (d), (e) and (f) show the deviation from the mean for each measurement with (g) being the combination of all three for each configuration.

The results of a grid and circular pattern provide similar results despite their different approaches. Coke shows for both methods the most segregation while pellet the least. A slight increase of both sinter and pellet segregation is observed in the circular pattern compared to the grid approach.

Table 15 Segregation index for material segregation of the manual defined sections

Material	Segregation index
Coke	0.22
Sinter	0.17
Pellet	0.11

While obtaining a segregation index is informative, it does not convey the entire narrative. Analysing pixel counts in sections opens additional possibilities. For instance, examining the ratio between materials per section offers more insights into the mixture's segregation and aids in comprehending the measurements. Material amounts can be measured either in pixels or as ratios relative to each other. Furthermore, assessing the ratio of materials per section provides insights into the composition on a smaller scale. Measuring per section also offers information on unlabelled data, which may include background elements, unidentifiable particles on lower layers, or particles overlooked by the model.
An illustrative example of measuring material distribution is depicted in Figure 34. This graph displays the percentage of each material in the section along with the quantity of unlabelled data, represented as black. The percentage of material present is similar to the before determined concentrations for the SI. All sections are represented, with empty sections indicating an absence of materials. The colour scheme corresponds to the materials mentioned earlier, and the black bar signifies pixels not associated with a particle.



(a) (b)

Figure 34 Material concentration (a) particle masks coloured based on material (b) the concentration of each material per section. Coke - blue, pellet - red, sinter – green, black – unlabelled pixels

Upon examining material presence in the sections, a noteworthy observation is the significant amount of coke near the outside with limited pellet and sinter. The distribution of pellet is almost evenly spread out across the heap. A slight increase is seen in the inner sections. Despite the marginal concentration increase, the segregation index remains low as the particles are divided over many sections. Sinter exhibits more variation within this measurement with clear high and low concentrations present in Figure 34 (b). Additionally, the unlabelled data within crowded sections is minimal, indicating that the presented measurements offer reliable information with minimal data gaps.

4.2.2. Results: size segregation

The model handles each particle as a distinct entity, facilitating the extraction of data concerning individual particles. However, due to the irregular shapes of particles, there is no singular measurement that perfectly characterizes their size. Notably, the surface area and Feret diameter are employed as estimators [119]. The surface area corresponds to the area-equivalent diameter, representing the size of the particle. This entails transforming the particle's area into a circle with an equivalent surface area, where the circle's diameter is regarded as the area-equivalent diameter, as shown in Figure 35. Another method to determine the size of a particle is the Feret diameter. The Feret diameter measures the distance between two parallel planes that touch the particle, with the maximum Feret diameter representing the longest such distance shown in Figure 36. Additionally, an elongation (EL) ratio is derived from the Feret diameters, providing insights into the particle's shape and distinguishing between long and thin versus more equal distributions such as squares or circles.

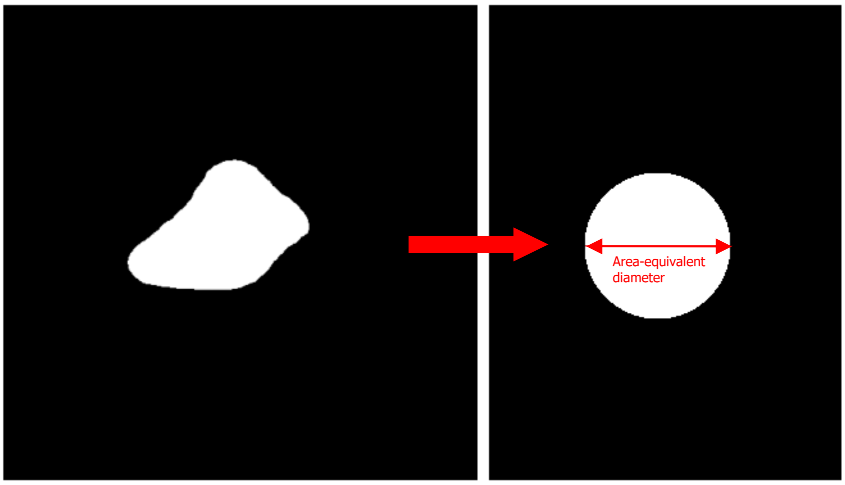


Figure 35 From particle to area-equivalent diameter

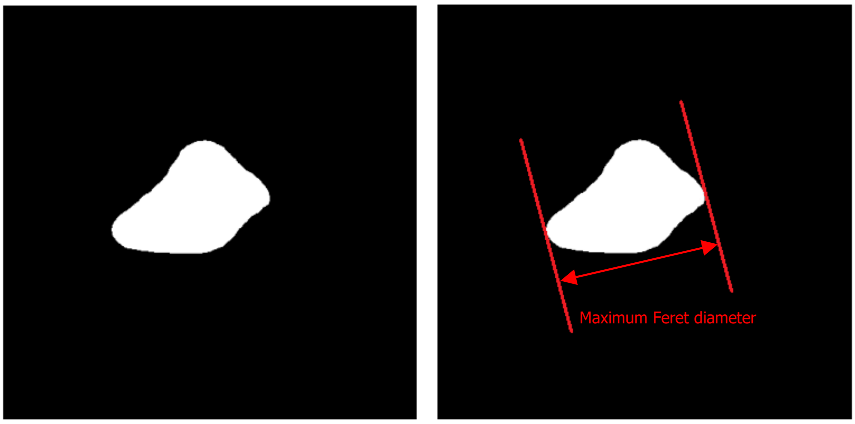
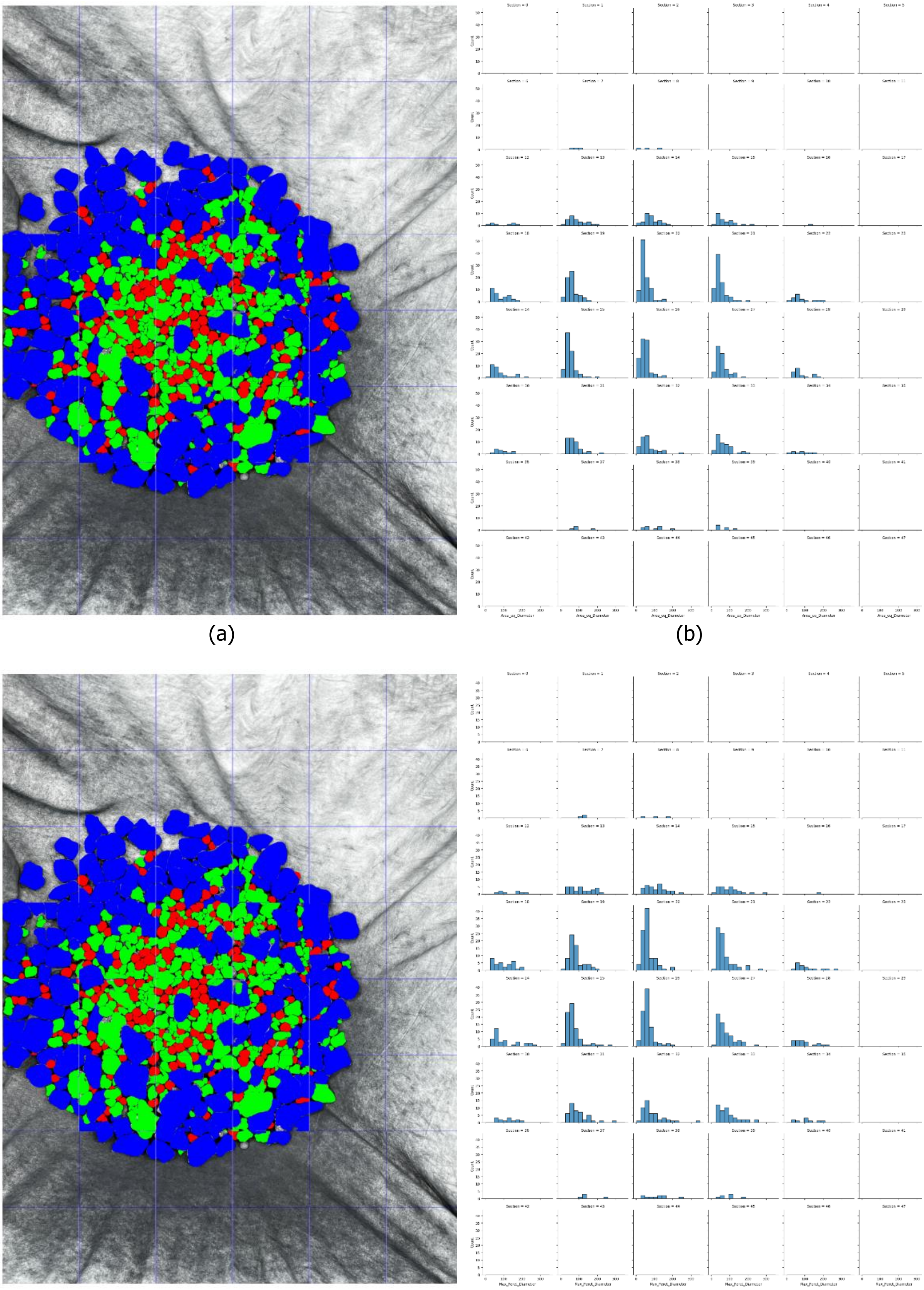


Figure 36 Particle shape and maximum Feret diameter

In the size measurements, it is important to consider that, due to image cropping, some particles may be fragmented into two to four pieces, creating additional particles for measurement. The model counted 857 particles, likely slightly higher than the actual count, as there are also particles partially obscured by others. Both separations in the image and overlapping particles influence the measured sizes.

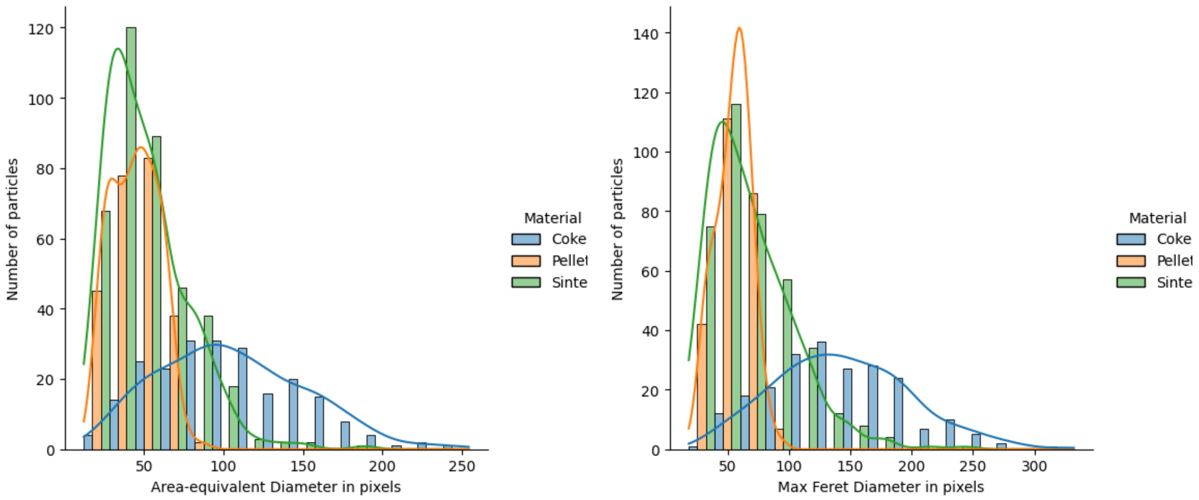


(a) (b)

(c) (d)

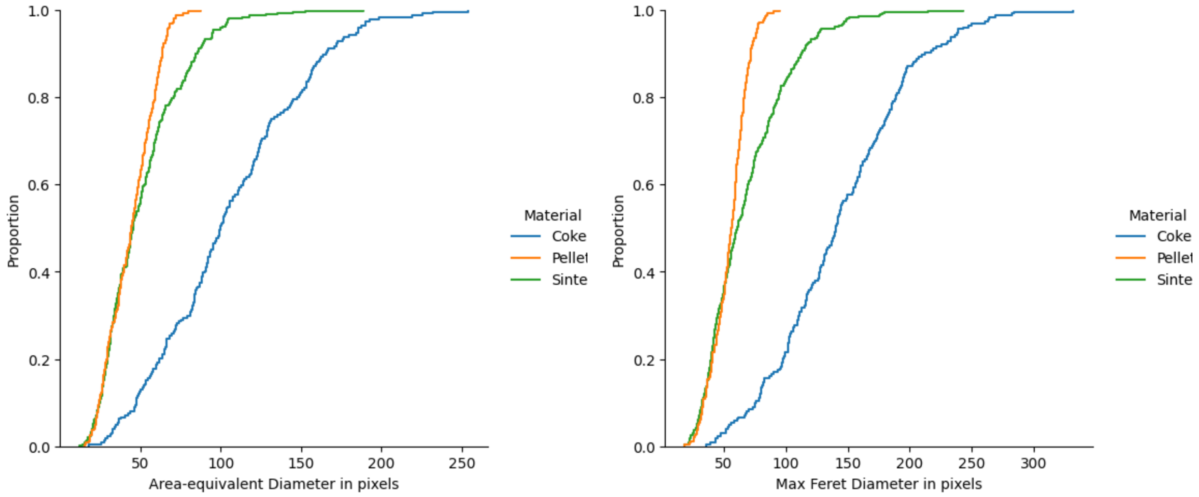
Figure 37 Size of particles per section of the image, (a) and (c) particle masks coloured per material (b) area-equivalent diameter (d) maximum Feret diameter

Figure 37 displays the particle size distribution per section, revealing a prominent peak in the number of particles for inner sections. In addition, the particle size distribution is able to be segmented based on materials, as presented in Figure 38. showcasing substantial differences due to measuring method in pellet particle diameters compared to coke and sinter. Instead of a narrow and high peak, pellet particles exhibit a smaller and wider peak with two tops, while coke and sinter demonstrate similar proportions in width and height. Moreover, the maximum size of measured particles is observed to be smaller when using the area-equivalent diameter, a distinction evident in the empirical distribution function (ECDF) plots as shown in Figure 39.



(a) (b)

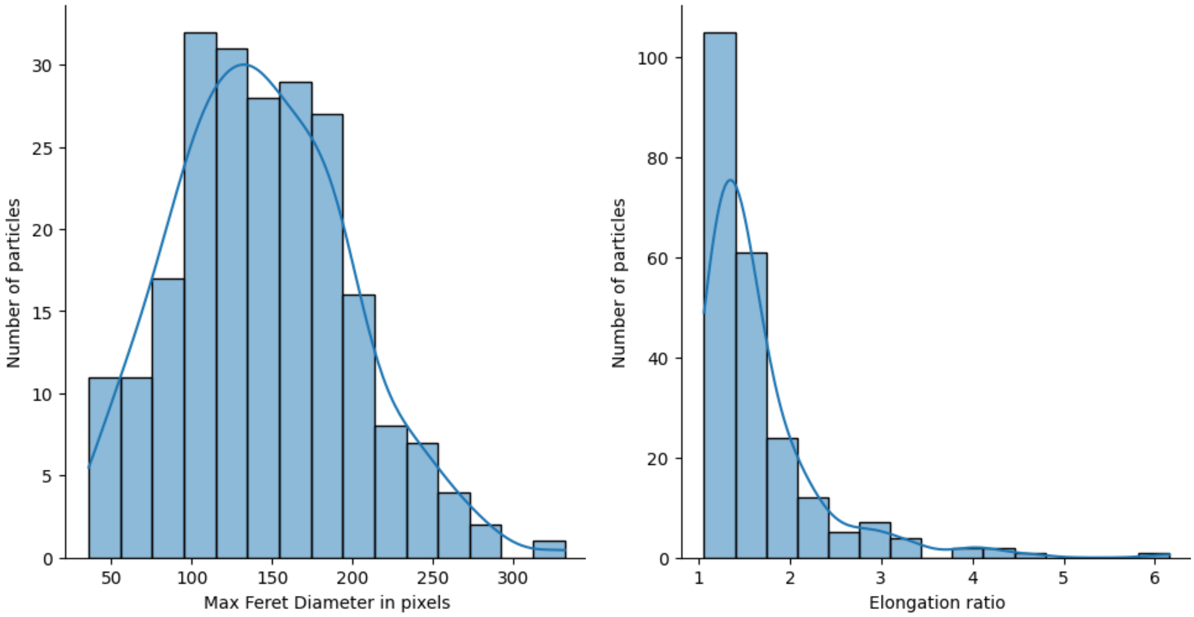
Figure 38 Size distribution graphs separated on material, (a) area-equivalent diameter (b) maximum Feret diameter



(a) (b)

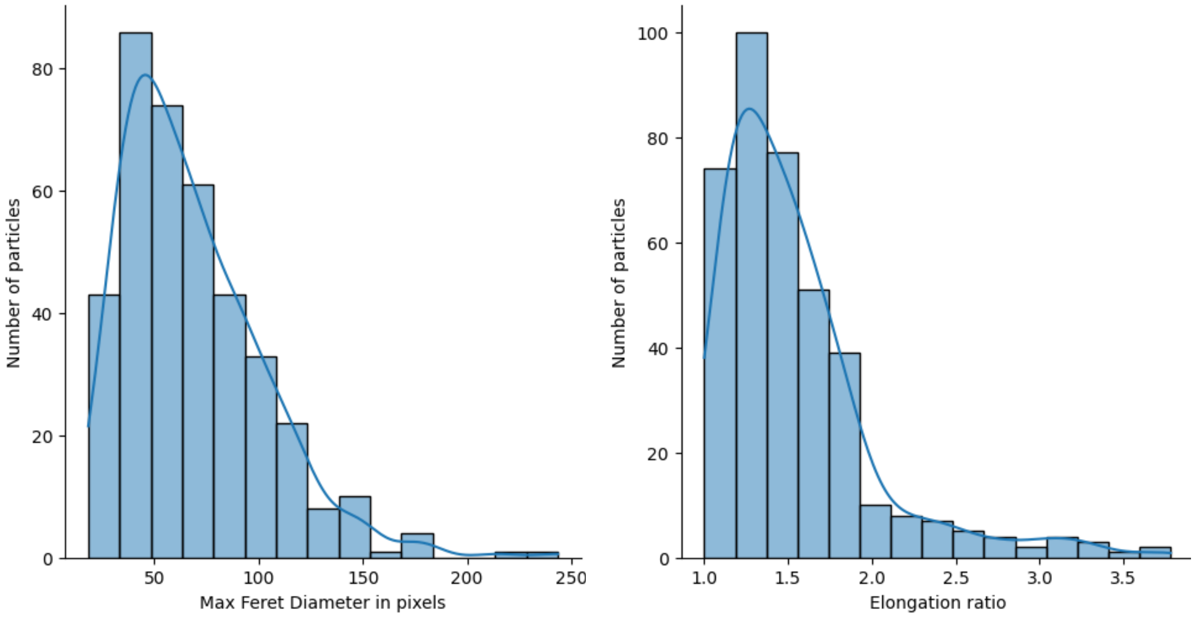
Figure 39 Size distribution ECDF graphs separated on material, (a) area-equivalent diameter (b) maximum Feret diameter

For analysing particle shapes, the elongation (EL) ratio is employed. The elongation signifies the aspect ratio between the particle's length and width. The ratio is derived from the maximum and perpendicular Feret diameters [119]. A value of 1 indicates a particle that can fit within a square box, resembling a square or round shape, while larger values correspond to more rectangular or oval shapes, indicative of longer and thinner particles. The subsequent graphs in Figure 40, Figure 41 and Figure 42 present the number of particles per maximum Feret diameter and EL for each material separately, offering insights into the size and shape of the materials.



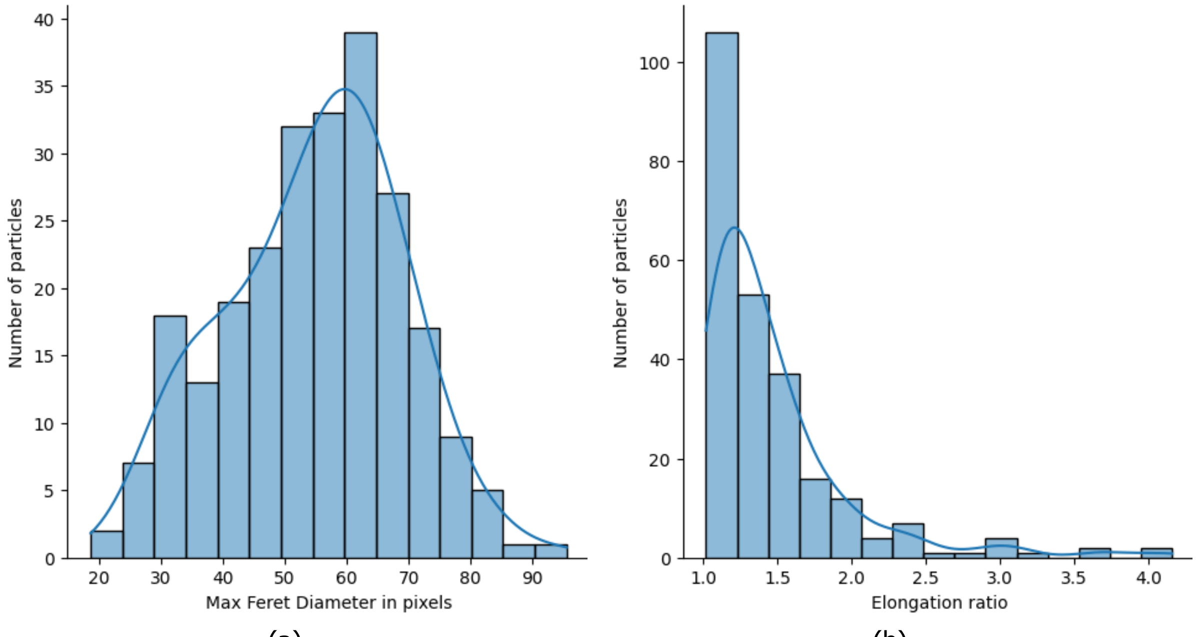
(a) (b)

Figure 40 (a) Number of particles per maximum Feret diameter (b) EL graph for coke particles



(a) (b)

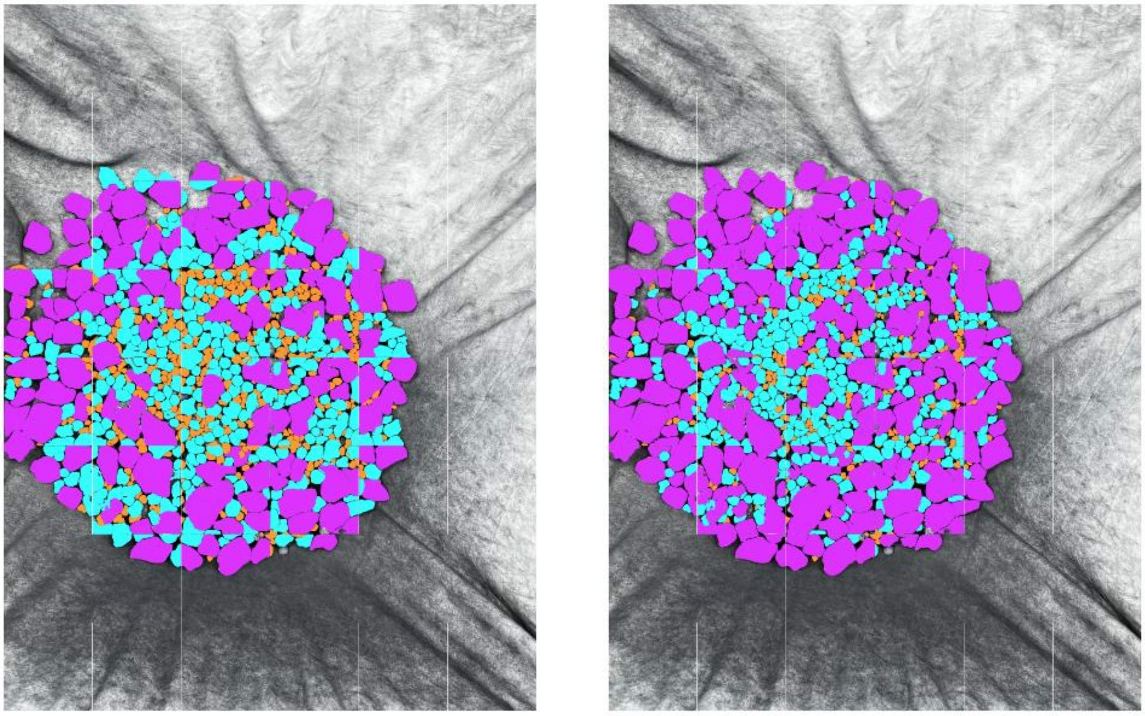
Figure 41 (a) Number of particles per maximum Feret diameter (b) EL graph for sinter particles



(a) (b)

Figure 42 (a) Number of particles per maximum Feret diameter (b) EL graph for pellet particles

Utilizing the information on particle size per section, a similar segregation index could be calculated. The particles are separated into categories of which their concentration in the sections is determined. This concentration is inserted into equation (1) to obtain the segregation indices presented in Table 16 and visualized in Figure 44. As seen in the distributions in Figure 38 and Figure 39, the maximum Feret diameter and the area equivalent diameter do not always give the same values for a particle. Therefore, both are presented. The diameter of the particles is measured with both methods and sorted based on size into three categories: smaller than 50 pixels, between 50 and 100 pixels, and larger than 100 pixels. These categories will go by the names small, medium, and large respectively. The categories are visualized in Figure 43, providing the masks with the colour based on their category. The diameter smaller than 50 pixels, between 50 and 100 pixels and larger than 100 pixels are coloured orange, light blue and purple, respectively. The difference between the two measuring methods is visible present as a significant number of particles are moved to different categories. The distribution of the 857 particles along the three categories is found in Table 17. These numbers additionally confirm the observation of the shift between categories for some particles when comparing both methods of measuring.



(a) (b)

Figure 43 Particle size masks, (a) area-equivalent diameter (b) maximum Feret diameter. Small – orange, medium – light blue, large - purple

Table 16 Segregation index for particle size segregation

Category	Segregation index	
	Max Feret	Area-equivalent
Small	0.03	0.06
Medium	0.14	0.13
Large	0.22	0.18

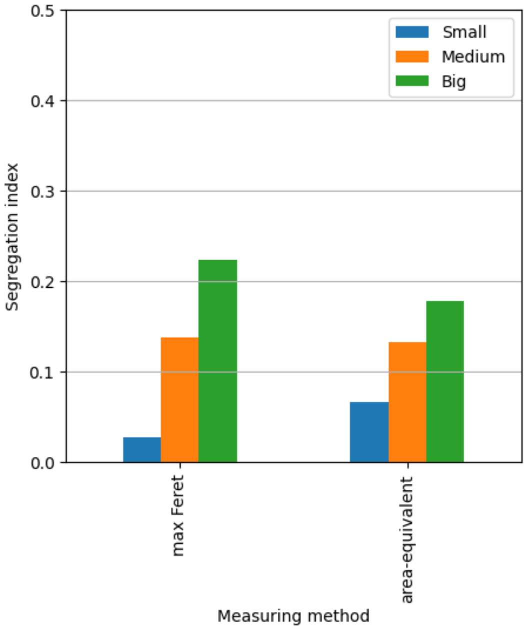


Figure 44 Segregation indices for the particle size categories per method, visualization of Table 16

Table 17 Number of particles per particle size

Category	Number of particles	
	Max Feret	Area-equivalent
Small	226	395
Medium	391	334
Large	240	128

4.2.3. Discussion

The model exhibits the capacity to assess segregation by leveraging equation (1), considering both material concentration within sections and particle size. Nevertheless, the accuracy of particle recognition significantly influences the measurements. Instances of incorrect recognition encompass missed particles, mislabelling non-particles, or erroneously labelled particles, resulting in underrepresentation or overrepresentation of components.

The partitioning of sections presents challenges as well, as it negatively impacts particle sizes and numbers for size segregation. Sections may inadvertently split medium-sized particles into two smaller ones, complicating recognition and analysis. A potential solution involves generating multiple images from the same setup with slight repositioning, mitigating the impact of section divisions. A mean from these images could reduce separation errors.

The need for separations arises from dataset constraints and model limitations, as not all image sizes can traverse the model. Creating a dataset with larger images compatible with the model could enhance performance but does not completely address the need for sections in segregation measurements.

Beyond structural errors imposed by grid-based image processing, model misinterpretations contribute to errors. Highlighted errors in Figure 45 include a false positive, where a cloth piece is misidentified as a coke particle, and false negatives, such as a missed pellet particle. Preventing these false positives could involve using higher contrasting colours for the background to enhance distinction. The missed pellet might be attributed to over-lighting, causing difficulties in recognition under unusual conditions.

Expanding the dataset with variations in lighting, achieved by image augmentation, presents a potential solution by exposing the model to more diverse scenarios.

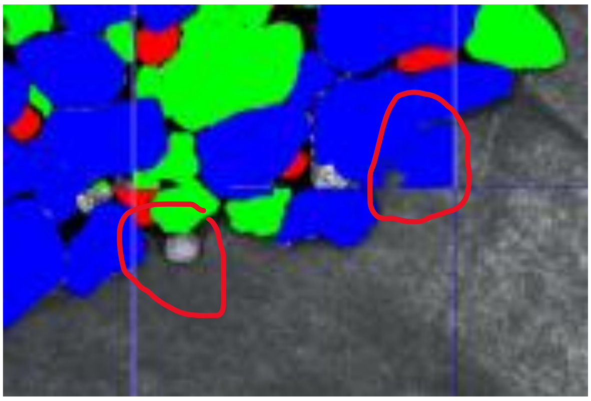


Figure 45 Highlight of two types errors in the prediction

4.3. Conclusion

The methodologies applied to image analysis can be extended to process the model outputs. As demonstrated the model is capable of differentiating particles, presenting them in a distinct fashion. Consequently, this enables the description of the mixture based on the materials and their spatial distribution, quantified through segregation indices. The processing of the image, conducted on a per-particle basis, empowers the model to incorporate other properties of the mixture, specifically the particle characteristics such as size and shape. The extraction of the material segregation is conducted in two manners, utilizing the preprocessing of the image for square shapes and the radial approach suggested for the heap structure. Between the two approaches minor deviations were observed

Collectively, these procedures provide a detailed understanding of the mixture's composition, offering insights into the size, shape, and material attributes of each particle. Despite the multitude of possibilities, the precision of these measurements is intricately linked to the model's performance. Instances of missing particles or mislabelling can introduce variability and exert influence on the accuracy of the results. Consequently, there is a heightened emphasis on the dataset, as it sets the conditions within which the measurements are conducted. The robustness of the dataset becomes essential in ensuring the reliability and accuracy of the obtained insights.

5. Conclusions

This chapter revisits the preceding section to address the main research question:

How can artificial intelligence be utilized to analyse and measure both material and particle size segregation in a granular mixture?

This question is explained through the exploration of the following sub-questions:

What is the state of the art for measuring and quantifying segregation in granular materials?

The current state of the art involves two distinct measurement approaches: invasive (sampling) and non-invasive (scanning). The latter predominantly utilizes optical measurements due to their equipment simplicity. However, optical methods confront limitations in differentiating particle categories, especially when similar groups are present. Consequently, preprocessing, such as painting or coating, is often used to enhance distinction. Moreover, stable lighting conditions are a necessity for optical measurements to mitigate colour leakage due to varying illumination. Quantifying segregation relies heavily on the concentration of particles and is often derived from standard deviation. With optical measurements utilizing specific formulas based on pixel concentration within image sections.

How can artificial intelligence be utilized through deep learning and computer vision to measure segregation in a granular mixture, and what are considerations for selecting a model and evaluating its performance?

For an AI system to consistently interpret images, an adaptable algorithm capable of identifying particles across varying positions, shapes, and lighting conditions is essential. Hence, a deeper investigation into deep learning is undertaken due to its elevated performance capabilities and superior generalization skills. Deep learning exhibits enhanced proficiency in recognizing particles amidst diverse lighting, shapes, and sizes, even in unexplored conditions. To derive meaningful interpretations from images, deep learning models necessitate training. This iterative process involves exposing the model to a set of images annotated with expected outcomes. Over successive iterations, the deep learning model attains the ability to replicate and transfer its acquired knowledge to novel images.

The learning mechanism of a deep learning model involves adjusting the weights in the neural network and filters. Filters dictate how the model perceives objects, while the neural network attributes labels to them. Determining optimal parameters is achieved through loss functions and the Gradient Descent algorithm. Subsequently, a trained model analyses input images, generating three essential types of information for each particle: category, bounding box, and mask. These outputs can be translated to material, location, and corresponding pixels for a given particle. Since the material is associated with groups of pixels, the segregation quantification formula used in optical measurements becomes applicable.

The most important factor is the dataset, both its quantity and quality. The efficacy of deep learning is dependent upon extensive datasets, enabling generalizations. The model's performance improves with exposure to a diverse array of conditions. Moreover, the quality of annotations within images assumes significance, with meticulous annotations setting a high standard for the predictions, while lax annotations may lead to loosely created masks. When drawing conclusions based on pixel count, the quality of masks becomes a crucial factor influencing the outcomes.

A model exerts a significant influence on performance. Various models exhibit varying proficiencies in learning or discerning patterns. To discern these disparities in performance, metrics can be employed for model comparisons. Among the various available metrics, precision and recall are the most commonly employed. These metrics quantify the accuracy of predictions and the capacity to identify all relevant instances, respectively. A frequently used derivative of these metrics is average precision, which is commonly applied for evaluation on standardized datasets such as MSCOCO, facilitating comparative analysis between models. It is important to note that performance metrics derived from standard datasets may not consistently correlate with performance on custom datasets. However, due to the lack of a direct correlation of model components and the performance. A comparison on the same dataset is the best indication for performance available.

The evaluation of model performance reveals two distinct trends. Firstly, the majority of high-performing models leverage Transformer architectures, whereas more widely adopted models primarily employ conventional neural networks. Informed by comparisons presented in literature utilizing standardized datasets, a top-performing model was selected: ISTR, which is based on the Transformer architecture. In addition to this high-performing model, the commonly utilized Mask R-CNN was included for comparison. Furthermore, Cascade Mask R-CNN, which has significantly influenced the evolution of modern models and shares similar methodologies with ISTR, was also incorporated. This model, built upon the Mask R-CNN, implements an iterative improvement process to enhance performance, similar to strategies observed in transformer-based models.

How to select the most suitable artificial intelligence for measuring segregation in a granular mixture and what insights can the artificial intelligence provide on the composition and segregation of the mixture?

From the comparison test of three models on our dataset, the model Cascade Mask R-CNN was selected. While the performance based on numerical values was slightly lower than Mask R-CNN, the processed output made clear it performed better than the values suggested. Labelling more particles and with more prediction correct than its competitors. Therefore, Cascade Mask R-CNN was used to investigate the potential information to be abstracted with this method of measuring.

Three outputs are generated by the model which present a multitude of potential applications. Primarily, the mixture can be segregated based on the materials identified, thereby facilitating the determination of material segregation through the utilization of a segregation index. Additionally, an in-depth analysis of material distribution allows for a more nuanced understanding of the segregation index.

As the mixture undergoes processing on a particle-by-particle basis, several supplementary avenues for information extraction become apparent. Particle size can be ascertained through metrics such as the Feret diameter or the area-equivalent diameter. Armed with knowledge about the size of all particles, it becomes possible to explore the size distribution within the mixture. Furthermore, similar to material segregation, size segregation can be determined by assessing the distribution of particle sizes. The final feature under consideration relates to the shape of particles. Specifically, the elongation ratio. A metric that encapsulates the shape characteristics of particles within the mixture.

To summarize this thesis, the results demonstrate a significant potential in the acquisition of information concerning both the mixture and its individual particles. The outputs facilitate a comprehensive analysis that presents greater opportunities compared to traditional methodologies. The employed artificial intelligence algorithms exhibit a strong capability for effective particle differentiation. By identifying intricate patterns, AI can recognize particles without necessitating preprocessing or consistent lighting. This advancement surpasses the limitations associated with traditional measurement techniques and substantially streamlines the measurement process. This approach would serve as a compelling solution for industries where traditional methods are impractical or for applications involving special conditions, such as elevated temperatures or highly sensitive materials.

With the three output types, category, bounding box and masks, the AI generates multiple perspectives and insights regarding mixtures. The detailed information extracted for each individual particle allows for comprehensive characterization of the mixture in terms of material composition, size, shape, and their respective distributions, as well as both material and size segregation.

Crucially, the dataset is the primary driver of the performance, as well as the scalability of the AI. A broader diversity of particles during the training phase enables the algorithm to be applied across a wider range of mixtures. This results in the development of a unified tool capable of measuring any of the included particle types, which can be expanded as necessary.

6. Recommendations

This project introduces deep learning methodologies to particulate science, specifically focusing on the measurement of segregation. However, acknowledging the nuanced nature of this challenge, the current model's performance could be further optimized, and its applicability expanded by introducing diverse materials or extracting additional information. This section delves into opportunities for extending and utilizing this project as a foundation for future developments.

1. **Dataset considerations:**
   - **Interchangeable:** The dataset plays a critical role in the model's abilities, providing the foundation for learning particle recognition. One key aspect is the interchangeability of datasets among different models, facilitating ease of transferability or updates. This is exemplified in this thesis by training three different models with the same dataset. While the format of annotations may differ based on the model and task, standardizing representations, such as using COCO format, enables flexibility.
   - **Impact on model performance:** Despite the saying "quality over quantity," in the realm of deep learning, both are crucial. The dataset's quality, in terms of precise annotations, coupled with its size, contributes to the model's robustness and accuracy.

2. **Dataset expansion:**
   - **Increasing dataset size:** Deep learning's effectiveness correlates with extensive data. Enlarging the dataset would enhance model performance, making it more adept at identifying materials and reducing errors caused by lighting or noise, as illustrated in Figure 45. A larger dataset contributes to more accurate measurements.
   - **Incorporating new materials:** Introducing new materials into the dataset expands the model's capabilities. Sufficiently representing diverse particles enables the model to detect and distinguish novel materials, broadening the range of detectable components within a mixture.

3. **Dataset creation challenges:**
   - **Manual annotation:** The creation of datasets demands substantial resources, with manual inputs necessary for supervised learning. Depending on the computer vision task (classification, object detection, or segmentation), annotations may involve labelling images, marking objects with bounding boxes, or delineating precise shapes for segmentation. Precision in annotations significantly impacts model performance.
   - **Automated annotation:** The quick expansion of datasets can be facilitated through the utilization of a pre-trained model for generating annotations. Such an approach requires the availability of a model proficient in particle recognition. This methodology produces the swift expansion of the dataset, leveraging the model's inherent capability to discern particles. Additionally, the generated images can undergo processing via augmentation tools, thereby simulating diverse conditions and consequently strengthening the model's robustness.

4. **Training process options:**
   - **Starting from scratch vs. continued training:** When incorporating additional data, two training approaches exist. Starting from nothing is recommended for substantial dataset changes, ensuring maximal utilization of new data. Alternatively, continuing training from a pre-existing model with new data accelerates the training process, requiring less time for increased performance.
   - **Hyperparameter optimalisation:** The training process consists of a large number of parameters influencing the learning of the model. These parameters can be optimized for improved learning rates, recognition and decision-making. Therefore, an improved version can be made without introducing new or altering components.

5. **Transfer learning:**
   - **Utilizing pretrained models:** Leveraging pretrained models expedites training, as they continue upon knowledge from similar tasks [120]. For instance, a model trained on coke, sinter, and pellet particles can be applied to a project involving different granular materials, saving training time and resources.

6. **Realistic environment integration:**
   - **Background and lighting conditions:** Introducing realistic environmental elements, such as background features and varied lighting conditions, would enhance the model's adaptability.

Simulating extreme lighting effects and colour tints corresponding to material properties could contribute to a more authentic representation.

7. **Speed considerations:**
   - **Optimizing measurement speed:** Considering the time constraints for measurements, an exploration of optimizing the processing speed is warranted. Focusing solely on classification and bounding boxes, or employing particle count as an indicator of size, could expedite the measurement process.

8. **AI integration with traditional methods:**
   - **Binary presentation of masks:** Since masks can be presented in a binary manner, AI recognition could replace conventional image processing, while traditional methods and algorithms handle subsequent image analysis. This approach allows for seamless integration into complex systems.

Conclusion:

In conclusion, numerous prospects exist for expanding upon this project. Most of these opportunities lie in the dataset, emphasizing the need for both quality and quantity. Expanding the dataset in a diversified manner and augmenting it with varied scenarios can reinforce the performance and enable operation in a broader spectrum of conditions. Considerations include the addition of more materials to recognize and the possibility of faster-paced environments. Furthermore, the project could be proposed as a replacement for the image processing step, while retaining traditional means of image analysis through the output of black and white material images. The project provide potential for the creation of a tool that is able to recognize a variety of materials that is easily applicable for experiments once it is setup. Providing opportunities to measure in previously unthinkable environments.

7. Implementation framework for continuation

This chapter provides a detailed examination of the practical aspects involved in establishing an AI project, explained in eight steps. These steps outline critical considerations for initiating the project from the ground up. Follow up iterations of the project can utilize the established works for continuation, allowing for modifications to the existing project parameters or datasets to align with the requirements of the new project.

- A project with a similar objective but utilizing different or additional materials can be duplicated, although modifications to the dataset are necessary. This involves adjustments to Step 3, followed by a continuation from Step 5.
- Changing the model necessitates additional steps to ensure proper functionality, as the model is depending upon a specific operational environment and data format. This process would involve starting from Step 2, while the dataset creation from Step 3 can be bypassed. However, it is crucial that the dataset annotations are exported in a format compatible with the new model.
- In instances where the task is altered, optimal performance is likely achieved by beginning from the initial stage. Given that the existing project is created for segmentation, other tasks can be executed at a more gradual pace compared to model designed for the task. However, this adjustment only requires a revaluation of the outputs to extract the sought information.

Step 1: task for AI

The initial step involves determining the specific information required from the AI. This refers to the three categories of computer vision models: classification, object detection and localization, and segmentation. The AI's output must correspond with the project's objectives. Relevant literature or existing projects can serve as references for potential implementations. Opting for the simplest model type facilitates easier dataset creation, accelerates the learning process, and enhances processing efficiency upon completion. Nonetheless, this choice may hinder the integration of more complex model capabilities. For example, segmentation encompasses both segmentation and object detection and localization, whereas object detection and localization do not include segmentation functionalities. The flowchart depicted in Figure 46 streamlines the decision-making process regarding model selection to expedite this determination.

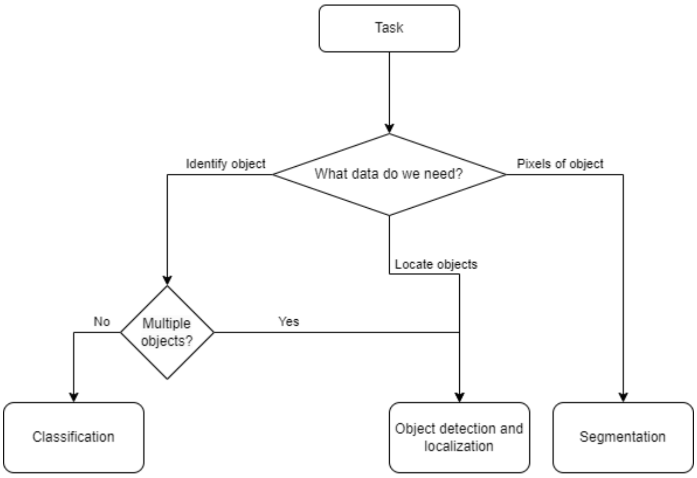


Figure 46 Flowchart computer vision model type selection

Step 2: Choosing model

A model must be selected for the project based on the type identified in step 1. Provided that the model type aligns, all such models should be capable of completing the specified task. Variations among models primarily concern to their processing speed and accuracy. Various sources are available for obtaining a model. Many models are accessible on GitHub, frequently referenced in their associated research publications. An alternative and more approachable option for newcomers is to utilize models available in libraries such as MMlabs [121] and Detectron2 [122]. These libraries provide a comprehensive catalogue of models that can be easily implemented. An additional advantage of utilizing these libraries is the active communities associated with them. Similar to widely adopted models, a larger user base facilitates troubleshooting and support when encountering difficulties.

Step 3: Dataset and annotations

In order for the model to effectively comprehend the task requirements, a considerable volume of examples must be provided. These examples are referred to as a dataset, which comprises of images and their corresponding annotations. To enhance the probability of accurate object recognition by the model, it is essential that the images present the objects within contextual environments. Given the existence of various annotation formats, it is imperative to export annotations in the appropriate format. The dataset must be constructed in alignment with the specific task and model being utilized; the correct labels must be assigned to the objects in the images. For the model to learn to identify an object, it must encounter the object recurrently across diverse settings, positions, and configurations. A lack of variability in the object's presentation will hinder the model's ability to detect variations of that object. Furthermore, the dataset should present the multiple instances of the objects across numerous images. While there is no predetermined quantity for the required appearances or images, practice suggests that an increased number of instances correlates with improved performance [123].

To facilitate the image annotation process, several tools are available to improve the tediousness of this task. Section 4.1.1 presents various options for annotation tools. A wide selection is available, A tool that aligns with one's preferences can be used. It is crucial to note that the quality of the dataset directly influences the model's output.

Step 4: Setting up environment

Prior to training a model, it is recommended to establish an environment made for the model's dependencies. Given that models often rely on specific versions of libraries, failing to do so may lead to conflicts with other models or scripts. Commonly utilized tools for creating such environments include Anaconda [124] and Docker [125], while a virtual environment is an alternative option. Each model necessitates its own environment due to distinct dependencies. However, utilizing a library offers several models that share compatible dependencies. Furthermore, having an active community that works with the model or library can facilitate troubleshooting when issues arise. The specific dependencies, along with their requisite versions, are typically provided during the model or library installation process, often in the form of a requirements file that is validated during setup.

Many models can be executed using Python and C++, often as a combination of both languages. Consequently, having access to Python expands the pool of available models. The environment must have access to an NVIDIA GPU, as numerous models depend on CUDA [126] for training. While some models may function on a CPU, it is important to note that CPU processing is significantly slower than GPU processing for image handling. For instance, a task that takes 10 minutes on a GPU might require a couple of hours on a CPU, or tasks that take a few hours on a GPU are extended to several days on a CPU. As a result, utilizing a GPU is strongly advised. Furthermore, working with larger datasets demands an increased amount of VRAM. If local GPUs are unavailable, alternative options include leveraging a virtual environment with GPU capabilities, such as Google Colab [127].

For a full installation guide, it is advisable to check the specific documentation associated with the model or library, as installation procedures may vary. For instance, the installation of ISTR, as documented on their GitHub repository [128], requires specific versions of PyTorch [129], torchvision [130], and the CUDA toolkit; these versions may not correspond to the latest releases or be compatible with other installed versions. The command `git clone` is employed to retrieve the model from GitHub, while `cd` navigates to the directory of the cloned model. The final command initiates the installation process; when executed in a command window, the invocation of Python directs the system to use Python to run the setup script with the `build` and `develop` commands.

```
install pytorch==1.6.0 torchvision==0.7.0 cudatoolkit=10.1 -c pytorch
pip install opencv-python
pip install scipy
pip install shapely
git clone https://github.com/hujiecpp/ISTR.git
cd ISTR
python setup.py build develop
```

Step 5: Training

The training process is relatively straightforward. After configuring the environment and defining the hyperparameters, the training loop is executed for the predetermined number of iterations. Taking ISTR as an example, a single command is issued to initiate the training script. Similar to step 4, this operation is conducted in a command window, again necessitating the invocation of Python to execute the script. Specifically, the script train_net.py is employed, which calls upon the dataset, associated functions, and hyperparameter settings for the training process. The subsequent command-line inputs the number of usable GPUs for the image processing and the model for which the weights will be optimized during training.

```
python projects/ISTR/train_net.py --num-gpus 4 --config-file projects/ISTR/configs/ISTR-R50-3x.yaml
```

Step 6: Performance evaluation

To evaluate the success of the training process, two primary methods can be employed: analysing performance metrics or conducting a practical test of the model. Based on the observer's assessment, a decision is to be made to either continue training for additional iterations, adjusting the hyperparameters and restart the training process, or to finalize the model.

For testing the model, it is advisable to utilize images that were not included in the training set to obtain the most accurate assessment of the learning progress. This can be accomplished using a separate test dataset or by providing the model with individual images and manually assessing the predictions. While the latter method may lack precision, it offers a quick means of gauging the model's current state. While employing a dedicated dataset is more advantageous for tracking progress across multiple training iterations.

Step 7: Saving model

The model at this stage is divided into two components: the model structure and the associated weights. This model is operations with the separation while allowing for switching out the weights or continuing the training process at later stages. However, both components can be merged into a single unit for enhanced integration, resulting in a more efficient and faster processing tool.

Model exportation can be performed to various formats to ensure compatibility across different applications and systems. A significant advantage of this approach is the reduction in the overall storage size of the model. Maintaining a separation between the model and the weights can require tens of gigabytes of storage, which is suboptimal for transferring the model or the implementation within other applications. The supported export formats include ONNX [131], NCNN [132], TensorRT [133], PyTorch and Keras [134] frameworks. Each having their expertise and compatibilities.

Step 8: Output exploration

Upon inputting an image, the model generates outputs specific to the model type utilized. Each model type produces distinct outputs, which can be examined to extract relevant information. The data is structured in a tensor comprising multiple layers. For segmentation tasks, the outputs consist of three components: class, bounding box, and mask. The class indicates the type of object detected, while the bounding box provides the coordinates representing the object's location within the image. The mask is represented as a binary image, matching the dimensions of the input image, with pixels corresponding to detected objects marked as ones. Each detected object in the image is assigned a unique index within these three components. Therefore, element i in the class is the same object as element i in the list of bounding boxes and masks.

Similar to traditional methods, the data derived from the model requires processing. The outputs can be set to match expectations from traditional techniques, facilitating the transition from legacy programs or scripts to AI-based approaches. Nevertheless, thorough exploration of the generated data may produce new and valuable insights, as the AI is able to separate individual particles and components within the image.

For processing using separated weights and a model for ISTR, the following command is used. This command invokes a Python script titled "demo" for image processing, accompanied by the following parameters: the model configuration file, the input image to be analysed, the designated output data location, the confidence threshold for predictions, and the file path for the weight parameters.

```
python demo/demo.py --config-file projects/ISTR/configs/ISTR-R50-3x.yaml --input input1.jpg --output ./output --confidence-threshold 0.4 --opts MODEL.WEIGHTS ./output/model_final.pth
```

Final remark

The eight steps provide a broad outline for considerations when initiating an AI project. However, variability may arise when implementing different models. Consequently, it is advisable for novices in this area to adhere to the instructions associated with the models and avoid lesser-known alternatives. This is needed, as the probability of successfully setting up a model on the initial attempt is low, even when following the prescribed guidelines.

References

[1] OECD, "Plastics use in 2019," 2022. [Online]. Available: https://doi.org/10.1787/efff24eb-en.

[2] M. Garside, "Statista," 29 8 2023. [Online]. Available: www.statista.com/statistics/264775.

[3] M. Garside, "Statista," 20 2 2023. [Online]. Available: www.statista.com/statistics/1224214.

[4] H. Ritchie, P. Rosado and M. Roser, "Agricultural production," *Our World in Data,* 2023.

[5] J. Yoon, "Application of experimental design and optimization to PFC model calibration in uniaxial compression simulation," *International Journal of Rock Mechanics & Mining Sciences ,* pp. 71-9, 2007.

[6] X.-J. Lui, Y.-j. Zhang, X. Li, Z.-f. Zhang, H.-y. Li, R. Liu and S.-j. Chen, "Prediction for permeability index of blast furnace based on VMD–PSO–BP model," *Journal of Iron and Steel Research International,* vol. 31, pp. 573-583, 2024.

[7] Y. Yu and H. Saxén, "Experimental and DEM study of segregation of ternary size particles in a blast furnace top bunker model," *Chemical Engineering Science,* vol. 65, no. 18, pp. 5237-5250, 2010.

[8] X. F. Dong, A. Jayasekara, D. Sert, R. Ferreira, P. Gardin, S. J. Chew, D. Pinson, B. J. Monaghan and P. Zulli, "Slag Flow in the Packed Bed With Varied Properties and Bed Conditions: Numerical Investigation," *Metallurgical and Materials Transactions B,* vol. 54, pp. 56-69, 2023.

[9] A. D. Rosata and D. L. Blackmore, "IUTAM Symposium on Segregation in Granular Flows," in *IUTAM Symposium on Segregation in Granular Flows*, Cape May, 1999.

[10] J. M. Ottino and D. V. Khakhar, "Mixing and segregation of granular materials," *Annual review of fluid mechanics,* vol. 32, no. 1, pp. 55-91, 2000.

[11] A. Hadi, R. Roeplal, Y. Pang and D. L. Schott, "DEM Modelling of Segregation in Granular Materials: A Review," *KONA Powder and Particle Journal,* vol. 41, pp. 78-107, 2014.

[12] F. J. Muzzio, P. Robinson, C. Wightman and D. Brone, "Sampling practices in powder blending," *International journal of pharmaceutics,* vol. 155, no. 2, pp. 153-178, 1997.

[13] X. Liu, C. Zhang and J. Zhan, "Quantitative comparison of image analysis methods for particle mixing in rotary drums," *Powder Technology,* vol. 282, pp. 32-36, 2015.

[14] P. Gajjar, C. G. Johnson, J. Carr, K. Chrispeels, J. M. N. T. Gray and P. J. Withers, "Size segregation of irregular granular materials captured by time-resolved 3D imaging," *Scientific Reports,* vol. 11, p. 8352, 2021.

[15] J. Gray and C. Ancey, "Multi-component particle-size segregation in shallow granular avalanches," *Journal of Fluid Mechanics,* vol. 678, pp. 535-588, 2011.

[16] D. N. Mondal, W. Lui, H. Bartusch, Y. Kaymak, T. Paananen, O. Mattila and H. Saxén, "Numerical Study of Gas Flow and Temperature Patterns in the Blast Furnace Throat," *Metallurgical and Materials Transactions B,* vol. 53, pp. 3882-3895, 2022.

[17] J. A. de Castro, C. Takano and J.-i. Yagi, "A theoretical study using the multiphase numerical simulation technique for effective use of H2 as blast furnaces fuel," *Journal of Materials Research and Technology,* vol. 6, no. 3, pp. 258-270, 2017.

[18] C. C. Xu and D.-q. CANG, "A Brief Overview of Low CO2 Emission Technologies for Iron and Steel Making," *Journal of Iron and Steel Research, International,* vol. 17, no. 3, pp. 1-7, 2010.

[19] R. Weinekötter, "Mixing of solid materials," in *Production, Handling and Characterization of Particulate Materials*, Springer Cham, 2015, p. 291–326.

[20] J. C. Williams, "The segregation of particulate materials. A review," *Powder Technology,* vol. 15, no. 2, pp. 245-251, 1976.

[21] S. R. de Silva, A. Dyrøy and G. G. Enstad, "Segregation Mechanisms and Their Quantification Using Segregation Testers," *IUTAM Symposium on Segregation in Granular Flows. Solid Mechanics and Its Applications,* vol. 81, pp. 11-29, 2000.

[22] A.-N. Huang and H.-P. Kuo, "Developments in the tools for investigation of mixing in particulate systems - A review," *Advanced Powder Technology,* vol. 25, no. 1, pp. 163-173, 2014.

[23] P. Porion, N. Sommier, A.-M. Faugere and P. Evesque, "Dynamics of size segregation and mixing of granular materials in a 3D-blender by NMR imaging investigation," *Powder Technology,* vol. 141, no. 1-2, pp. 55-68, 2004.

[24] K. M. Hill, Y. Fan, J. Zhang, C. Van Niekerk, E. Zastrow, S. C. Hagness and J. T. Bernhard, "Granular segregation studies for the development of a radar-based three-dimensional sensing system," *Granular Matter,* vol. 12, pp. 201-207, 2010.

[25] D. M. Koller, A. Posch, G. Hörl, C. Voura, N. Urbanetz, S. Radl, S. D. Fraser, W. Tritthart, F. Reiter, M. Schlingmann and J. G. Khinast, "Continuous quantitative monitoring of powder mixing dynamics by near-infrared spectroscopy," *Powder Technology,* vol. 205, no. 1-3, pp. 87-96, 2011.

[26] J. M. Prats-Montalbán, A. d. Juan and A. Ferrer, "Multivariate image analysis: A review with applications," *Chemometrics and Intelligent Laboratory Systems,* vol. 107, no. 1, pp. 1-23, 2011.

[27] J. W. Woods, "Chapter 7 - Image Enhancement and Analysis (Second Edition)," *Multidimensional Signal, Image, and Video Processing and Coding,* pp. 223-256, 2012.

[28] M. Asachi, E. Nourafkan and A. Hassanpour, "A review of current techniques for the evaluation of powder mixing," *Advanced Powder Technology,* vol. 29, no. 7, pp. 1525-1549, 2018.

[29] S. Jiang, X. Long, Y. Ye, J. Lui, S. Yang, X. Xiao and S. Lui, "Experimental research on the motion behavior of particles in a rotating drum based on similarity," *Powder Technology,* vol. 398, p. 117046, 2022.

[30] C.-Y. Hung and C. P. Stark, "Enhanced-g centrifuge experiments of granular flows in a drum," University of Minnesota, 2014. [Online]. Available: https://sedexp.net/experiment/enhanced-g-centrifuge-experiments-granular-flows-drum. [Accessed 18 02 2024].

[31] M. Poux, P. Fayolle, J. Bertrand, D. Bridoux and J. Bousquet, "Powder mixing: Some practival rules applied to agitated systems," *Powder Technology,* vol. 68, pp. 213-234, 1991.

[32] P. V. Danckwerts, "The definition and measurement of some characterists of mxtures," *Applied Scientific Research,* vol. Section A 3, pp. 179-296, 1952.

[33] D. V. Khakhar, J. J. McCarthy and J. M. Ottino, "Radial segregation of granular mixtures in rotating cylinders," *Physics of Fluids,* vol. 9, no. 12, pp. 3600-3614, 1997.

[34] H. Li and J. J. McCarthy, "Cohesive particle mixing and segregation under shear," *Powder Technology,* vol. 164, no. 1, pp. 58-64, 2006.

[35] D. Shi, A. A. Abatan, W. L. Vargas and J. J. McCarthy, "Eliminating segregation in free-surface flows of particles," *Physical Review Letters,* vol. 99, no. 14, p. 148001, 2007.

[36] C.-C. Liao, H.-W. Lan and S.-S. Hsiau, "Density-induced granular segregation in a slurry rotating drum," *International Journal of Multiphase Flow,* vol. 84, pp. 1-8, 2016.

[37] C. C. Liao, S. S. Hsiau and K. To, "Granular dynamics of a slurry in a rotating drum".

[38] S. H. Chou, C. C. Liao and S. S. Hsiau, "An experimental study on the effect of liquid content and viscosity on particle segregation in a rotating drum," *Powder Technology,* vol. 201, no. 3, pp. 266-272, 2010.

[39] P. Lacey, "Developments in the theory of particle mixing," *Journal of Applied Chemistry,* no. 4, pp. 257-268, 1954.

[40] J. Beaudry, "Blender efficiency," *Chemical Engineering,* no. 55, p. 112, 1948.

[41] F. Valentin, "Mixing of powders and pasts-some basic concepts," *Transactions of the Institution of Chemical Engineers and the Chemical Engineer,* no. 45, p. CE99, 1967.

[42] T. Sakaino, "On the Degree of Mixing and the Mixing Procedure for Powdered Materials," *Journal of the Ceramic Association, Japan,* vol. 65, p. 171, 1958.

[43] C. Legatt, "Newsletter," *Association of Official Seed Analysts,* vol. 25, p. 3, 1951.

[44] A. Lastovtev, M. Khyalnova and J. Makarov, *Khim. Prom. SSSR,* vol. 11, p. 815, 1962.

[45] M. Elgendy, Deep Learning for Vision Systems, Shelter Island, NY: Manning Publications Co., 2020.

[46] IBM, "What is artificial intelligence (AI)?," IBM, 2023. [Online]. Available: https://www.ibm.com/topics/artificial-intelligence. [Accessed 28 12 2023].

[47] J. A. Bullinaria, *IAI : The Roots, Goals and Sub-fields of AI,* Birmingham: University of Birmingham, 2005.

[48] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development,* vol. 3, no. 3, pp. 210-229, 1959.

[49] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[50] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778.

[51] C. Janiesch, P. Zschech and K. Heinrich, "Machine learning and deep learning," *Electron Markets,* vol. 31, pp. 685-695, 2021.

[52] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science,* vol. 2, p. 420, 2021.

[53] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[54] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Leaning Deep features for Discriminative Localization," in *IEEE conference on computer vision and pattern recognition*, 2016.

[55] M. Kuhn and K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, Taylor & Francis Group, 2019.

[56] A. G. Ganie and S. Dadvandipour, "From big data to smart data: a sample gradient descent approach for machine learning," *Journal of Big Data,* vol. 10, no. 162, 2023.

[57] U. B. Trivedi, M. Bhatt and P. Srivastava, "Prevent Overfitting Problem in Machine Learning: A Case Focus on Linear Regression and Logistics Regression," *Innovations in Information and Communication Technologies,* pp. 345-349, 2020.

[58] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan and J. Walsh, "Advances in Computer Vision," in *Computer Vision Conference (CVC 2019)*, Las Vegas, 2020.

[59] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham and L. A. Gabralla, "A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions," *IEEE Access,* vol. 10, pp. 107293-107329, 2022.

[60] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohaer, M. Rokonuzzaman and I. Hossain, "A comprehensive review on autonomous navigation," *arXiv preprint arXiv:2212.12808,* 2022.

[61] J. Chai, H. Zeng, A. Li and E. W. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications,* vol. 6, p. 100134, 2021.

[62] Y. Chen, Y. Tian and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," *Computer Vision and Image Understanding,* vol. 192, no. 1077-3142, p. 102897, 2020.

[63] Y. Zhou, Y. Ren, E. Xu, S. Liu and L. Zhou, "Supervised semantic segmentation based on deep learning: a survey," *Multimedia Tools and Applications,* vol. 81, no. 20, p. 29283–29304, 2022.

[64] Y. Chen, M. Mancini, X. Zhu and Z. Akata, "Semi-Supervised and Unsupervised Deep Visual Learning: A Survey," *IEEE transactions on pattern analysis and machine intelligence,* 2022.

[65] N. Le, R. Singh, K. Yamazaki, K. Luu and M. Savvides, "Deep Reinforcement Learning in Computer Vision: A Comprehensive Survey," *Artificial Intelligence Review,* pp. 1-87, 2022.

[66] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen and I. Sutskever, "Zero-Shot Text-to-Image Generation," *International conference on machine learning,* pp. 8821-8831, 2021.

[67] V. K. Garg and A. Kalai, "Supervised unsupervised learning," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montréal, Canada, 2018.

[68] N. Zeng, H. Li, Z. Wang, W. Lui, S. Lui, F. E. Alsaadi and X. Lui, "Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip," *Neurocomputing,* vol. 425, pp. 173-180, 2021.

[69] S. Indolia, A. K. Goswami, S. P. Mishra and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Computer Science,* vol. 132, pp. 679-688, 2018.

[70] Random Stuff, "Space Lion Triangle," 2012. [Online]. Available: https://www.zazzle.com/store/freebirdstriangle. [Accessed 2024].

[71] X. Ding, X. Zhang, J. Han and G. Ding, "Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, IEEE, 2022, pp. 11953-11965.

[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[73] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud and V. Shet, "Multi-digit number recognition from Street View imagery using deep convolutional neural networks," in *International Conference on Learning Representations*, 2014.

[74] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[75] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," in *IEEE conference on computer vision and pattern recognition*, 2017.

[76] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision (ICCV),* pp. 2980-2988, 2017.

[77] T. Sercu and V. Goel, "Dense Prediction on Sequences with Time-Dilated Convolutions for Speech Recognition," in *30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

[78] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu and Y. Wei, "Deformable Convolutional Networks," *Computer Vision and Pattern Recognition,* 2017.

[79] Y. Li, N. Miao, L. Ma, F. Shuang and X. Huang, "Transformer for object detection: Review and benchmark," *Engineering Applications of Artificial Intelligence,* vol. 126, p. 107021, 2023.

[80] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR,* 2021.

[81] H. Ceasar, J. Uijlings and V. Ferrari, "COCO-Stuff: Thing and Stuff Classes in Context," *Computer Vision and Pattern Recognition (CVPR),* pp. 1209-1218, 2018.

[82] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International journal of computer vision,* vol. 115, pp. 211-252, 2015.

[83] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso and A. Torralba, "Semantic Understanding of Scenes through the ADE20K Dataset," *International Journal of Computer Vision,* vol. 127, pp. 302-321, 2019.

[84] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 3213-3223, 2016.

[85] PaperswithCode, "Papers with Code," [Online]. Available: www.paperswithcode.com. [Accessed 9 22 2023].

[86] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, IEEE, 2018, pp. 6154-6162.

[87] S. Jung, H. Heo, S. Park, S.-U. Jung and K. Lee, "Benchmarking Deep Learning Models for Instance Segmentation," *Applied Science,* vol. 12, no. 17, 2022.

[88] Q. Yang, J. Peng and D. Chen, "A Review of Research on Instance Segmentation Based on Deep Learning," in *Proceedings of the 13th International Conference on Computer Engineering and Networks. CENet 2023. Lecture Notes in Electrical Engineering, vol 1126*, Singapore, 2024.

[89] R. Zhang, Z. Tian, C. Shen, M. You and Y. Yan, "Mask Encoding for Single Shot Instance Segmentation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 10226-10235, 2020.

[90] X. Wang, R. Zhang, T. Kong, L. Li and C. Shen, "SOLOv2: Dynamic and Fast Instance Segmentation," in *Neural Information Processing Systems*, Vancouver, Canada, 2020.

[91] D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, "YOLACT: Real-time Instance Segmentation," *Proceedings of the IEEE/CVF international conference on computer vision,* pp. 9157-9166, 2019.

[92] Y. Wang, Z. Xu, H. Shen, C. Bsoshan and L. Yang, "CenterMask: single shot instance segmentation with point representation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 9313-9321, 2020.

[93] R. Sharma, m. Saqib, C. T. Lin and M. Blumenstein, "A Survey on Object Instance Segmentation," *SN Computer Science,* vol. 3, p. 499, 2022.

[94] J. Hu, L. Cao, Y. Lu, S. Zhang, Y. Wang, K. Li, F. Huang, L. Shao and R. Ji, "ISTR: End-to-End Instance Segmentation with Transformers," *arXiv:2105.00637,* 2021.

[95] E. S. Marquez, J. S. Hare and M. Niranjan, "Deep Cascade Learning," *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS,* vol. 29, no. 11, pp. 5475-5485, 2018.

[96] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Lui, J. Shi, W. Ouyang, C. C. Loy and D. Lin, "Hybrid task cascade for instance segmentation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* p. 4974–4983, 2019.

[97] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *Proceedings of the IEEE/CVF international conference on computer vision,* pp. 10012-10022, 2021.

[98] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems,* vol. 28, 2015.

[99] X. Zhang, "Understanding Mask R-CNN Basic Architecture," Shuffle, 14 November 2021. [Online]. Available: https://shuffleai.blog/blog/Understanding_Mask_R-CNN_Basic_Architecture.html. [Accessed 01 March 2024].

[100] X. Wu, D. Sahoo and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing,* vol. 396, pp. 39-64, 2020.

[101] K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker and M. Z. Afzal, "Cascade Network with Deformable Composite Backbone for Formula Detection in Scanned Document Images," *Applied Sciences,* vol. 11, no. 16, p. 7610, 2021.

[102] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[103] A. Kulkarni, D. Chong and F. A. Batarseh, "5 - Foundations of data imbalance and solutions for a data democracy," in *Data Democracy*, Academic Press, 2020, pp. 83-106.

[104] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghain, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, pp. 658-666.

[105] J. V. Hurtado and A. Valada, "Chapter 12 - Semantic scene segmentation for robotics," in *Deep Learning for Robot Perception and Cognition*, Academic Press, 2022, pp. 279-311.

[106] P. Bruce, A. Bruce and P. Gedeck, Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python, Sebastopol, CA, USA: O'REILLY, 2017.

[107] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater,* 2007.

[108] N. Chinchor, "MUC-4 Evaluation Metrics," in *the 4th conference on Message understanding (MUC4 '92)*, McLean, Virginia, 1992.

[109] CVAT.ai Corporation, "Computer Vision Annotation Tool (CVAT)," CVAT.ai Corporation, 11 2023. [Online]. Available: https://github.com/opencv/cvat.

[110] M. Tkachenko, M. Malyuk, A. Holmanyuk and N. Liubimov, "Label Studio: Data labeling software," 2020-2023. [Online]. Available: https://github.com/heartexlabs/label-studio.

[111] A. Dutta and A. Zisserman, "The VIA Annotation Software for Images, Audio and Video," in *Proceedings of the 27th ACM International Conference on Multimedia*, New York, NY, USA, ACM, 2019, p. 4.

[112] "Labelbox," Labelbox, 2024. [Online]. Available: https://labelbox.com.

[113] "V7labs," V7labs, 2024. [Online]. Available: https://www.v7labs.com.

[114] B. Dwyer and J. Nelson, "Roboflow," Roboflow, 2022. [Online]. Available: https://roboflow.com.

[115] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017.

[116] F. Küppers, A. Haselhoff, J. Kronenberger and J. Schneider, "Confidence Calibration for Object Detection and Segmentation," *Deep Neural Networks and Data for Automated Driving,* p. 225–250, 2022.

[117] S. Wenkel, K. Alhazmi, T. Liiv, S. Alrshoud and M. Simon, "Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation," *Sensors (Basel, Switzerland),* vol. 21, no. 13, p. 4350, 2021.

[118] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterhiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[119] H. G. Merkus, Particle Size Measurements, Dordrecht: Springer, 2010.

[120] D. Vasani, "How do pre-trained models work?," *Towards Data Science,* 2019.

[121] K. Chen, J. Wang, J. Pang, Y. Xiong, X. Li, S. Sun, Cao, Yuhang, W. Feng, Z. Lui, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy and D. Lin, "MMDetection: Open MMLab Detection Toolbox and Benchmark," *arXiv preprint arXiv:1906.07155,* 2019.

[122] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo and R. Girshick, *Detectron2,* https://github.com/facebookresearch/detectron2, 2019.

[123] C. Sun, A. Shrivastava, S. Singh and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," in *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[124] Anaconda Software Distribution, "Distribution," Anaconda, [Online]. Available: https://www.anaconda.com/download. [Accessed 07 07 2024].

[125] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal,* vol. 2014, no. 239, p. 2, 2014.

[126] NVIDIA, P. Vingelmann and F. H. Fitzek, *CUDA,* https://developer.nvidia.com/cuda-toolkit, 2007.

[127] Google, "Google Colaboratory," [Online]. Available: https://colab.research.google.com/. [Accessed 04 August 2024].

[128] J. Hu, Y. Lu, S. Zhang and L. Cao, "hujiecpp ISTR," Github, 18 April 2024. [Online]. Available: https://github.com/hujiecpp/ISTR. [Accessed 25 July 2024].

[129] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jian, M. Voznesensky, B. Boa, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu and S. Chintala, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilatio," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, 2024.

[130] TorchVision maintainers and contributors, "TorchVision: PyTorch's Computer Vision library," *GitHub repository,* 11 November 2016.

[131] ONNX Runtime developers, *ONNX Runtime,* https://onnxruntime.ai/, 2021.

[132] H. Ni, *ncnn,* https://github.com/Tencent/ncnn, 2017.

[133] NVIDIA, "TensorRT Open Source Software," NVIDIA, 2024. [Online]. Available: https://github.com/NVIDIA/TensorRT. [Accessed 4 August 2024].

[134] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io. [Accessed 4 August 2024].

Appendix A: Scientific Research Paper

Segregation measurements in granular mixture with AI

1st Wessel de Jonge
Transport Engineering and Logistics
Technical University Delft
Delft, Netherlanders

2nd Ahmed Hadi
Transport Engineering and Logistics
Technical University Delft
Delft, Netherlanders

3rd Yusong Pang
Transport Engineering and Logistics
Technical University Delft
Delft, Netherlanders

4th Dingena Schott
Transport Engineering and Logistics
Technical University Delft
Delft, Netherlanders

Abstract— **Segregation is a significant factor that can affect the uniformity of a mixture. In order to address and mitigate segregation, accurate data is necessary to characterize particles throughout the process. However, obtaining data in a manner that does not impact the mixture can be challenging, particularly in non-laboratory settings where conditions may not be as controlled.**

In laboratory settings, stable lighting and coloured particles can be used to aid in differentiation. However, this approach may not be feasible for all materials. Therefore, the development of a tool that can identify particles without the need for colouring or consistent lighting is highly valuable.

This study focuses on a common mixture found in blast furnaces, consisting of coke, pellet, and sinter. The similarities in colour schemes and overlapping sizes of coke and sinter present challenges for particle recognition. Unstable lighting further complicates the differentiation process.

The proposed method is applied to a mixture containing all three components to demonstrate its capabilities. This method effectively distinguishes between the various particles, providing information on both the material and area of each particle. The measurement data is utilized to assess material and size segregation within the mixture. Material segregation is evaluated using a square grid and ring configuration, while size segregation is determined by categorizing particles into three groups based on diameter, including Feret diameter and area-equivalent diameter. Code and pre-trained models are available at https://github.com/WesselJonge/SMAI.

Keywords—Particle, Granular materials, Size segregation, Material segregation, AI, Deep learning

I. Introduction

During the manufacturing processes involving multiple component mixtures, control over the ratios is essential to meet product specifications. However, the natural phenomenon of segregation is counteracting both the mixing and the ability to maintain the mixed state [1]. As the particle-based industries are vast and are found in diverse fields, segregation is a common issue. Showing influence on the end products and efficiency of the manufacturing. A well-known example is the blast furnaces in the iron and steel industries. Within the blast furnaces, the permeability of the bed has a great influence on the efficiency of the process [2]. An efficient permeability for the bed is achieved with appropriate distribution of the large and fine particles on the burden surface. Segregation influences this distribution in undesired manners. Leading to negative effects on the permeability. Providing inconsistent pressure drops over the burden and inefficient usage of the gas. The consequences are both economic and environmental [3]. The segregation in a blast furnace is one of the many cases that highlight the importance of improving the understanding of the causes of segregation. Especially since segregation is a complex process influenced by numerous variables with diverse roots, as there are lots of contributing factors to the segregation [1] The dependencies on the material properties and the environment further complicate the phenomenon. For the material properties, the difference between the properties of the particles such as the size, shape and density but also the chemical affinity, moisture absorbability and magnetic properties are considered. Environmental influences range from weather conditions to system-specific factors in the form of surface roughness, vibrations or transport modes. The gain insight into segregation, experiments are conducted to recreate the process. Experimental setups allow for a controlled environment to help reduce external influences. Making it possible to take measurements and identification of segregation causes. Experiments with granular materials often focus on the variation between particles such as density or size difference [4].

Despite segregation being around for decades and being observed in common applications, extracting data on the composition of granular mixtures is not a trivial task [5]. There are two distinct approaches for extracting segregation measurements, either through intrusive or non-intrusive methods. Intrusive methods involve extracting a sample of the mixture for detailed component analysis [6]. While sampling is straightforward, it is destructive to the structure of the mixture. Which introduces errors in the measurement and following measurements. On the other hand, non-intrusive methods employ a wide array of techniques such as optics [7] or x-ray [8] waves. Collecting information without direct interaction with the mixture and therefore preserving the undisturbed state for accurate measurements [6].

Optical measuring provides a cost-effective and straightforward approach to quantifying segregation, which makes it highly appealing for applications beyond specialized laboratory investigations [5]. However, optics is reliant on image processing to facilitate the analysis. Requiring conditions with high colour contrast among particles and minimal light interference. To attain an adequate colour contrast may pose challenges when the materials exhibit similar colours. This issue is mitigated by painting or coating the particles to improve the differentiation based on colour. However, this solution is not always feasible as not all particles allow to be painted. Additionally, this is only suitable for setups in laboratories. For the lighting problem, a similar issue is seen. As appropriate lighting is difficult to achieve, fully enclosed setups are used to minimize light interference. These two issues with taking measurements mainly lay with the recognition of the particles. Therefore, this paper explores the options of utilizing AI for the recognition of the particles.

II. Segregation quantification

The segregation is quantified with indices. The indices describe the variation in the presence of the particles across a mixture. A similar quantification is done when measuring the degree of mixing. Despite both aiming for the opposite side of the spectrum, they measure the same spread of the particles [9]. For measuring segregation using optics, the formula often applied is the standard variation [10] [11] [12] [13] [14] [15] [16], in the form of:

$$SI = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \quad (1)$$

With N is the number of sections, x is the concentration in section i and $\bar{x}$ is the arithmetic mean of the concentration of all sections (2) [9].

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2)$$

The segregation index presented in (1) outputs a number between 0 and 0.5. Where 0 is perfectly mixed and 0.5 is completely segregated. The equation takes only one component into consideration for the quantification. The values index is determined based on the deviation from the mean presence of particles per section. Therefore, allowing the use to investigate mixtures of more than two components.

III. Utilizing AI for measuring

The recognition of inconsistently shaped and placed objects requires a form of flexibility from the AI. The flexibility is found in machine learning models that learn to detect the object. However, as the mixture presents a challenge due to the limited characterizing features of the particles, a statistical approach will be difficult. With this challenge, the machine learning models that utilize larger neural networks become interesting [17]. Machine learning with large neural networks is referred to as deep learning. Deep learning models utilize their large neural network to find the objects in addition to recognizing them. This makes the model able to use not directly visible patterns to distinguish the particles.

A. Computer vision

All AI models that interpret the environment and respond accordingly are considered to be part of computer vision [18]. Despite it being a broad term, computer vision primarily deals with visual inputs such as images and videos. There are several tasks a model is able to perform on an image. The main three tasks are classification, object identification and localization, and segmentation. Classification looks at the entire presented image and attempts to classify what is in it. Identification and localization looks for patterns in the image for objects. The objects that are found are then classified. Lastly, segmentation continues by marking the pixels corresponding to the object. The pixels marked are referred to as the mask. For the purpose of measuring the segregation, segmentation is required to accurately measure the particles' volume. Within segmentation, there are three distinct directions, instance, semantic and panoptic [19]. Instance segmentation keeps all segmented particles separate. While semantic segregation groups all particles that are from the same material in a single output. Panoptic segmentation marks every pixel in an image and links it to an object, regardless of it being a foreground object or background. To facilitate size segregation measurements all particles are needed to be presented one by one with the known material. Therefore, among the segmentation methods instance segmentation is the most suited for the application.

B. AI for image segmentation

Most of the models for segmentation make use of a convolution neural network (CNN) as they excel at handling image data and classification [20]. The CNN consist of layers of filters to process the image. The filters allow to highlight features in an image, making it easier to find patterns that signify the presence of an object. After several layers of filter, the image enters a neural network for the classification and mask generation.

The learning capabilities of a model are correlated with the number of parameters present in the model [21]. The parameters are spread throughout the model. There are parameters in the filters of the CNN, which alter the view on the image. Other parameters are found within the neural networks to signify the importance of inputs for the decision-making process. As the parameters have a large influence on the ability to learn and identify objects, models with a larger number of parameters have an increased performance.

IV. Method

In the field of image segmentation tasks, there is a wide range of models available for selection. Three potential models have been carefully chosen based on their relevance, popularity, and potential for success. Due to the lack of universally accepted metrics for evaluating model performance, the community has adopted alternative methods to standardize this process.

A common approach is to evaluate model performance using a standardized dataset, such as the popular MS COCO dataset. By assessing each model's learning capabilities with the same inputs, we can more accurately compare their performance. However, it is important to recognize that performance on one dataset may not necessarily indicate performance on another dataset. While this method allows us to assess a model's learning abilities, it may not fully capture its performance on all tasks.

For a more precise comparison of models for a specific task, it is crucial to test them using representative data and tasks.

A. Models of interest

The three models for further testing are Mask R-CNN [22], Cascade Mask R-CNN [23] and ISTR [24]. Mask R-CNN is one of the most popular segmentation model available. Mask R-CNN is the most basic model [25] and is even considered to be a classic for segmentation tasks [26]. Even though the model comes from 2017, its ongoing relevance is visible through the inclusion in performance comparison tables for newer models and the derivations created from Mask R-CNN. One of the derivations is Cascade Mask R-CNN [23]. Cascade Mask R-CNN introduces Cascade learning [27] to Mask R-CNN. By implementing refinement stages, the predictions are iterated to enhance the accuracy and improve the performance. When looking at the recent top-performing models on the standardized datasets, transformer-based models are shown to take the top spots. In the survey paper for Sharma [28], the transformer-based model ISTR shows potential on the standardized dataset by having the highest score. Transformers make use of combining features and position embedding to find objects. Transformers are often found in natural language processing (NLP) [29]. Similar to Cascade Mask R-CNN, transformers use a refinement feedback loop to improve the findings, boosting the accuracy and the performance of the model. However, a good score on the standardized dataset is an indication of their performance, but it might differ on another dataset.

Mask R-CNN consists of four key steps: the filters, a region proposal network, a classification network and the mask head. The filters prepare the images for the region proposal network. The region proposal network identifies potential objects on which the classification network and the mask head make their predictions. The classification network provides a class and a box capturing the object, a bounding box. The mask head provides the segmentation by identifying the precise shape of the object at the pixel level [22].

Cascade Mask R-CNN introduced a refinement loop into Mask R-CNN. The refinement is done on the classification network to narrow down the object more accurately [30]. After the refinement loops, the segmentation is done of the refined prediction of the object and is outputted together with the class and bounding box.

ISTR uses a different approach than Mask R-CNN and Cascade Mask R-CNN by utilizing a transformer for the predictions. The transformer takes filtered images and vectorizes the images by embedding the features with a position which are inputted into the self-attention module. [31] The self-attention module encapsulates complex relationships among different features. Additionally, a dynamic attention head is included to fuse RoI and the image features for the prediction head for the predictions. The predictions are made similarly to Mask R-CNN with the exemption of the segmentation head. The segmentation is done with the embedding, therefore to reconstruct the mask a mask decoder is applied. The recurrent refinement is done with the predictions by repeatedly updating the prediction boxes. Which refines the predictions and allows for in parallel the processing of the classification and segmentation.

B. *Comparing the models on a granular mixture*

The comparison of the models is only possible once the models are set up. The setting up of the models requires creating a dataset with the expected outputs and the training of the model. After which the model is able to process new images to detect and segmentate the particles in new situations. The application of the model is focused on the materials found in a blast furnace coke, sinter, and pellet. Therefore, a mixture of these three materials is used to set the model up. Therefore, the model is able to recognize the three materials once trained on the dataset. The comparison of the models is done by the performance metrics precision, indicating the correctness of the predictions, and recall, the ability to find all particles. These two performance indicators are then combined in a singular score to represent the performance with the F1-score [32], as in:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

Additionally, a visual evaluation is done to inspect the performance. Which allows the models to show their capabilities in an application case.

1) Dataset and training

The dataset consists of images that serve as examples for the models to learn from. Besides the images, an additional file is included telling the model the correct answers. The correct answers for what the model should detect are named annotations. For segmentation, the annotations consist of coordinates on the images. By linking the coordinates, a surface area is created which covers the object. During training, the predictions of the model will be compared with the created surface area and the difference is returned as feedback [33]. Based on the feedback the parameters within the model will be adjusted following the gradient descent algorithm [34].

The dataset created for the coke, sinter and pellet mixture is created in natural lighting and uses unprocessed particles. The used images are taken from a heap formed by releasing the mixture from a tube. A top-down view is used for the creation of the images. The dataset is divided into two sets of images. The first set will solely be used for training and will consist of 180 images. These 180 images contain around 13 thousand particles correlating to 6.5 thousand sinter, 2 thousand coke and 4.5 thousand pellet particles. The second set is smaller with 35 images and is used to measure the performance. The two sets of images allow for the use of cross-validation. Making the performance independent from the training and more representative of the end performance. The datasets are made with Roboflow [35], which provides various tools to make the annotating of the images more convenient. Tools include features such as altering the brightness or contrast of the image, to make the objects more visible for annotating. These changes in appearance are not saved onto the images in the dataset. The process of annotating consists of providing per particle in the image the material and the coordinates that describe the surface area. Which requires the manual selection of pixels in the image.

The training for all three models is set to make 40 thousand iterations, each iteration includes the feedback for changing the parameters. Which are roughly 222 epochs. The learning rate is set at 0.001 for both Mask R-CNN and Cascade Mask R-CNN, while ISTR required a far lower learning rate of 5e-5 to prevent divergences. The learning rate indicates the aggressiveness of the change in the parameters in the gradient descent algorithm. A larger learning rate uses bigger steps to find the best solution. Therefore, it is recommended to decrease the step size near the end of the training to get closer, than the larger step allows due to their size, to the optimal parameter values. Hence, the last four thousand steps have a reduced step size of a magnitude of 10 and the final 2 thousand are another tenfold smaller.

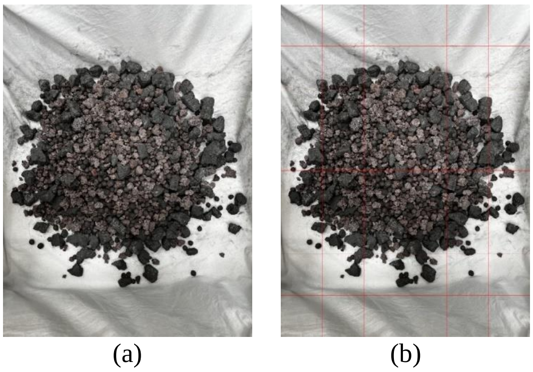

(a) (b)

Figure 3 Unused image in the training process utilized for the visual analysis, (a) image of the heap, (b) the division of the 48 sections

2) *Comparison of the models*

The evaluation set allows to extract the numerical performance from the model. Within the set, there are several images presenting the mixture in different arrangements. The scores represent the ability to recognize the different particles and correctly identify the material on the images in the evaluation set. As the model is selected to segment the objects, the scores are given at the pixel level. Therefore, being a few pixels off does influence the scoring of the performance. Despite the error being minimal and the possibility of the drawn annotation being slightly off due to the nature of connecting the coordinates not perfectly lining up with the particles. The resulting performance indices are shown in Table 1. The table provides next to the F1-score, also the precision in the form of the average precision (AP) and the recall as the average recall (AR). The average precision and average recall are taken due to the recognition and identification of multiple objects in several images.

Table 1 Scores of the trained deep learning models

	F1-score	**AP**	**AR**
Mask R-CNN	0.69	0.67	0.70
Cascade Mask R-CNN	0.66	0.63	0.69
ISTR	0.25	0.21	0.29

The visual evaluation consists of processing a larger image and looking at the detections by eye. To process a larger image, the image needs to be resized to fit the input and preference of the model. As the backbone uses a 224 x 224 resolution [36], the image is going to be resized for all sizes. The average size of the images used in the datasets is around 600 x 550. As the model is used to dimensions around that size, the model performs better when giving an image close to the average. An additional constraint to the size of the input image is the original size of the image. Therefore, the original image is cut into even sections to provide to the model for processing. The original image is cut into 48 sections, 6 horizontal by 8 vertical, with a resolution of 504 x 504. Resulting in the grid of images seen in Figure 3. All the 48 sections are individually inputted into the model and processed. To highlight the different materials recognized by the model, the pixels corresponding to the materials coke, sinter and pellet are labelled blue, green, and red respectively. After reassembling the sections, the images in Figure 1 with the materials highlighted are created. The second set of images in Figure 2 shows the predictions with a lower confidence threshold. The confidence threshold [37] filters out predictions for which the model assumes a low probability of being correct. Which allows to see the further potential or struggles of a model. As the models are permitted to show the particles of which the model is less certain.

C. *Decision*

Taking both the numerical and the visual evaluation into consideration, the model that is used for further analysis of the mixture is Cascade Mask R-CNN. In the numerical evaluation, Mask R-CNN and Cascade Mask R-CNN are close in scores. Mask R-CNN does seem to have the upper hand with the average precision. However, taking the visual evaluation into consideration, the roles are reversed with Cascade Mask R-CNN showing better performance. While a good score is indicative of performance, a number cannot always show the entire picture. As for the case of Mask R-CNN the struggle with larger sinter particles. ISTR showed the lowest performance of all three in both evaluations and therefore not considered to be used for further analysis of the mixture. ISTR does show promise with the lowered confidence threshold. Making it likely that ISTR would have needed more iterations to get to a similar performance as Mask R-CNN or Cascade Mask R-CNN. Transformers do not have the initial intuition on how to handle image data, unlike CNNs. Therefore, transformers require iterations to learn to handle the data. However, under set constraints, it was not able to catch up to the other two models.

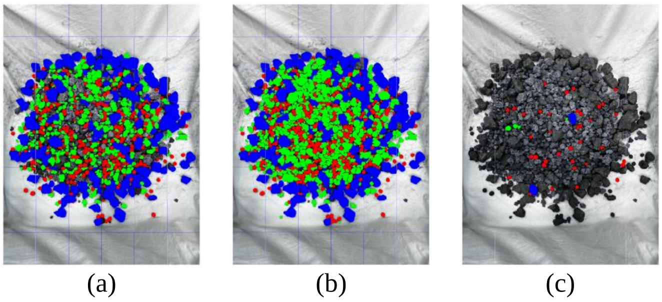

(a) (b) (c)

Figure 1 Predictions from the models (a) Mask R-CNN, (b) Cascade Mask R-CNN, and (c) ISTR
Coke - blue, pellet - red, sinter – green

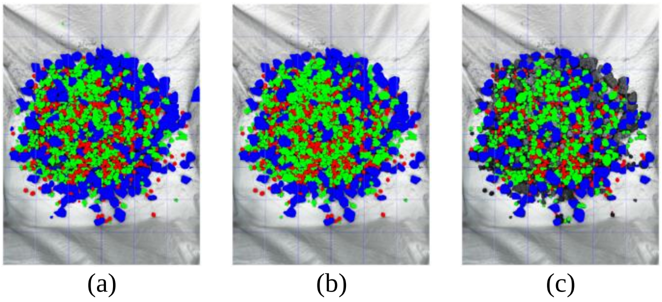

(a) (b) (c)

Figure 2 Predictions from the models with confidence threshold lowered (a) Mask R-CNN, (b) Cascade mask R-CNN, and (c) ISTR
Coke - blue, pellet - red, sinter – green

Figure 5 A single output of the segmentation model

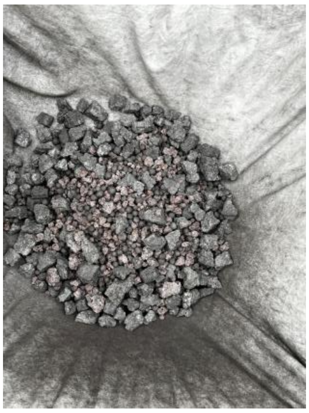
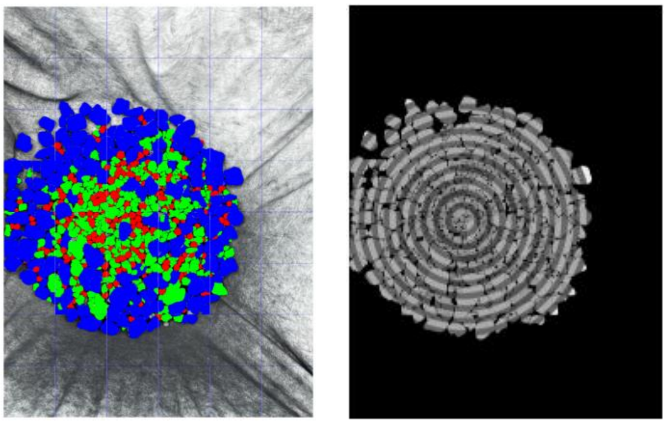


Figure 6 Unused image in the training process used for extraction of segregation measurements



(a) (b)

Figure 4 (a) Materials highlighted in the 48 sections configuration, (b) The selected ring configuration Coke - blue, pellet - red, sinter – green

V. Results

Both the material and size segregation are considered while measuring the segregation in the mixture. Due to the model processing per particle, a single particle can be isolated in the form of Figure 5 for measuring the size of the particle. Which enables the inclusion of size segregation. Similar to the visual evaluation an unused image, shown in Figure 6, in neither training nor evaluation, is cut up into sections and inputted into the model. The quantification of the segregation is done with (1) for all cases.

A. *Material segregation*

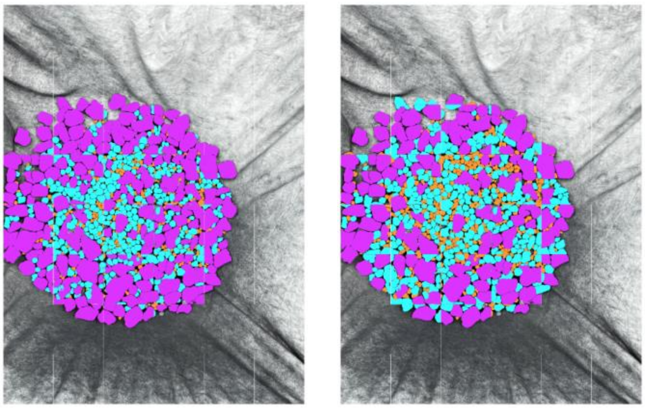
The material segregation is measured by identifying and separating the materials within the images. The different materials within the mixture are visualized in Figure 4a. Once the pixels and their corresponding material are known, the concentrations of the materials within defined areas are determined. The concentrations are determined with two different approaches. The first approach uses the predetermined sections from the input and the second approach uses a ring structure. Both approaches prescribe areas wherein the concentrations are calculated. The predetermined section does not allow for any tuning of the section. Therefore, an ample amount of background is included which provides low concentrations for those sections. The approach making use of the ring structure follows more of the shape of the heap. Reducing the number of empty pixels included in the equation. As the ring configuration uses custom-defined areas, a minor calibration process was done to select the combination for the number of rings and the diameters of the rings. For the calibration, the effect of the combinations on the SI is investigated. When the combinations have similar outmost diameters, the differences in the segregation measurements are minimal. However, amongst the combinations with similar outer diameter, there was one relatively large outlier which had a lower measurement compared to the other combinations. This combination consisted of the least number of rings, highlighting the importance of having enough areas to compare in the calculations. The differences between the combinations with a larger number of areas are minor and the values hover around similar values. The combination with the least deviation from the mean of the different combinations is used to extract the SI. The configuration consists of 19 rings with a 60 pixel ring diameter, as shown in Figure 4b. The comparison between the SI of the two approaches in Table 2 shows differences in the measurements for sinter and pellet with pellet having the largest difference.

Table 2 Material segregation index

	Segregation index	
Method	**Squares**	**Rings**
Coke	0.22	0.22
Sinter	0.15	0.17
Pellet	0.07	0.11

B. *Size segregation*

Since the model processes the image particle-by-particle, individual particles are isolated in the output. The isolated particles are measurable in pixels. For the particle size, the following two methods are used for measuring the size of the particles: the maximum Feret diameter and the Area-equivalent diameter [38]. The maximum Feret diameter



(a) (b)

Figure 7 Particle size masks, (a) maximum Feret diameter, (b) area-equivalent diameter.
Small – orange, medium – light blue, large - purple

places two parallel planes on opposite sides of the particle and determines the largest distance between the planes while planes both touch the particle. The area-equivalent diameter takes the surface area of the particle and converts the area into a circle with an equal-sized surface area of which the diameter is determined. Both methods are used to describe irregularly shaped particles.

The particles in the mixture are measured and divided into three categories, small, medium, and large, for both methods. These categories use the following particle size criteria: smaller than fifty pixels, between fifty and a hundred pixels, and larger than a hundred pixels respectively. Figure 7 shows the classification of the particle size applied to the mixture, which presents the mixture in a similar manner as the figures for the material segregation. The particles are coloured by using the following colour scheme: small – orange, medium – light blue and large - purple. Equation (1) is applied to quantify the size segregation for both methods and presented in Table 3. For the size segregation, only the square sections are used for the segregation index.

Table 3 Size segregation index

	Segregation index	
Method	**Feret**	**Area-equivalent**
Small	0.03	0.06
Medium	0.14	0.13
Large	0.22	0.18

VI. Discussion

The possibilities for the extracted data are not limited to the shown results. More detailed information is available on the mixture such as the concentration of materials per section to further understand the measured index. Additionally, the model allows counting the particles per material, size, within a section or a combination of the mentioned properties. Providing additional details on the composition of the mixture.

The results presented are based on the outputs of the same model. The SI values are determined using the same established method and equations. Precise recognition of the model is essential for accurate calculations. However, there is room for improvement in the model's outputs. In terms of material segregation, two primary types of errors are identified: missing particles and false identification of particles. The former leads to decreased concentration in a section, while the latter increases the concentration, both of which can impact the SI depending on the magnitude of the error. Similarly, size segregation faces challenges in recognition, with a more significant impact due to the division of particles at section boundaries.

A potential solution lies within the dataset itself. Errors such as missing particles and background markings stem from insufficient data and variability to differentiate particles from the background. By expanding the dataset with a greater number of images, the model can learn from a wider range of examples, ultimately enhancing its performance. For size segregation, linking sections to the model and aligning them with the expected input size from training data can be effective. Adjusting the size of training images to match the input size, without scaling the particles, can eliminate the need for sections.

VII. Conclusion

The proposed method for utilizing artificial intelligence in quantifying segregation within mixtures demonstrates promise. The presented case study involves a mixture that proves challenging for conventional methodologies. Through AI technology, the ability to identify various particles and materials is achieved, enabling simultaneous examination of both material and size segregation. The imaging setup lacked controlled lighting and contrasting coloured particles, facilitating measurements in diverse and uncontrolled environments. This opens up the possibility of conducting on-site measurements.

References

[1] A. D. Rosata and D. L. Blackmore, “IUTAM Symposium on Segregation in Granular Flows,” in *IUTAM Symposium on Segregation in Granular Flows,* Cape May, 1999.

[2] Y. Yu and H. Saxén, “Experimental and DEM study of segregation of ternary size particles in a blast furnace top bunker model,” *Chemical Engineering Science,* vol. 65, no. 18, pp. 5237-5250, 2010.

[3] X. F. Dong, A. Jayasekara, D. Sert, R. Ferreira, P. Gardin, S. J. Chew, D. Pinson, B. J. Monaghan and P. Zulli, “Slag Flow in the Packed Bed With Varied Properties and Bed Conditions: Numerical Investigation,” *Metallurgical and Materials Transactions B,* vol. 54, pp. 56-69, 2023.

[4] J. M. Ottino and D. V. Khakhar, “Mixing and segregation of granular materials,” *Annual review of fluid mechanics,* vol. 32, no. 1, pp. 55-91, 2000.

[5] A. Hadi, R. Roeplal, Y. Pang and D. L. Schott, “DEM Modelling of Segregation in Granular Materials: A Review,” *KONA Powder and Particle Journal,* vol. 41, pp. 78-107, 2014.

[6] F. J. Muzzio, P. Robinson, C. Wightman and D. Brone, “Sampling practices in powder blending,” *International journal of pharmaceutics,* vol. 155, no. 2, pp. 153-178, 1997.

[7] X. Liu, C. Zhang and J. Zhan, “Quantitative comparison of image analysis methods for particle mixing in rotary drums,” *Powder Technology,* vol. 282, pp. 32-36, 2015.

[8] P. Gajjar, C. G. Johnson, J. Carr, K. Chrispeels, J. M. N. T. Gray and P. J. Withers, “Size segregation of irregular granular materials captured by time-resolved 3D imaging,” *Scientific Reports,* vol. 11, p. 8352, 2021.

[9] M. Poux, P. Fayolle, J. Bertrand, D. Bridoux and J. Bousquet, “Powder mixing: Some practical rules applied to agitated systems,” *Powder Technology,* vol. 68, pp. 213-234, 1991.

[10] P. V. Danckwerts, “The definition and measurement of some characteristics of mixtures,” *Applied*

Scientific Research, vol. Section A 3, pp. 179-296, 1952.

[11] D. V. Khakhar, J. J. McCarthy and J. M. Ottino, "Radial segregation of granular mixtures in rotating cylinders," *Physics of Fluids,* vol. 9, no. 12, pp. 3600-3614, 1997.

[12] H. Li and J. J. McCarthy, "Cohesive particle mixing and segregation under shear," *Powder Technology,* vol. 164, no. 1, pp. 58-64, 2006.

[13] D. Shi, A. A. Abatan, W. L. Vargas and J. J. McCarthy, "Eliminating segregation in free-surface flows of particles," *Physical Review Letters,* vol. 99, no. 14, p. 148001, 2007.

[14] C.-C. Liao, H.-W. Lan and S.-S. Hsiau, "Density-induced granular segregation in a slurry rotating drum," *International Journal of Multiphase Flow,* vol. 84, pp. 1-8, 2016.

[15] C. C. Liao, S. S. Hsiau and K. To, "Granular dynamics of a slurry in a rotating drum".

[16] S. H. Chou, C. C. Liao and S. S. Hsiau, "An experimental study on the effect of liquid content and viscosity on particle segregation in a rotating drum," *Powder Technology,* vol. 201, no. 3, pp. 266-272, 2010.

[17] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[18] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan and J. Walsh, "Advances in Computer Vision," in *Computer Vision Conference (CVC 2019),* Las Vegas, 2020.

[19] A. Kirillov, K. He, R. Girshick, C. Rother and P. Dollár, "Panoptic Segmentation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 9404-9413, 2019.

[20] N. Le, R. Singh, K. Yamazaki, K. Luu and M. Savvides, "Deep Reinforcement Learning in Computer Vision: A Comprehensive Survey," *Artificial Intelligence Review,* pp. 1-87, 2022.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR,* 2015.

[22] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision (ICCV),* pp. 2980-2988, 2017.

[23] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, USA, IEEE, 2018, pp. 6154-6162.

[24] J. Hu, L. Cao, Y. Lu, S. Zhang, Y. Wang, K. Li, F. Huang, L. Shao and R. Ji, "ISTR: End-to-End Instance Segmentation with Transformers," *arXiv:2105.00637,* 2021.

[25] S. Jung, H. Heo, S. Park, S.-U. Jung and K. Lee, "Benchmarking Deep Learning Models for Instance Segmentation," *Applied Science,* vol. 12, no. 17, 2022.

[26] Q. Yang, J. Peng and D. Chen, "A Review of Research on Instance Segmentation Based on Deep Learning," in *Proceedings of the 13th International Conference on Computer Engineering and Networks. CENet 2023. Lecture Notes in Electrical Engineering, vol 1126,* Singapore, 2024.

[27] E. S. Marquez, J. S. Hare and M. Niranjan, "Deep Cascade Learning," *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS,* vol. 29, no. 11, pp. 5475-5485, 2018.

[28] R. Sharma, m. Saqib, C. T. Lin and M. Blumenstein, "A Survey on Object Instance Segmentation," *SN Computer Science,* vol. 3, p. 499, 2022.

[29] Y. Li, N. Miao, L. Ma, F. Shuang and X. Huang, "Transformer for object detection: Review and benchmark," *Engineering Applications of Artificial Intelligence,* vol. 126, p. 107021, 2023.

[30] X. Wu, D. Sahoo and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing,* vol. 396, pp. 39-64, 2020.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems,* 2017.

[32] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater,* 2007.

[33] M. Elgendy, Deep Learning for Vision Systems, Shelter Island, NY: Manning Pyblications Co., 2020.

[34] A. G. Ganie and S. Dadvandipour, "From big data to smart data: a sample gradient descent approach for machine learning," *Journal of Big Data,* vol. 10, no. 162, 2023.

[35] B. Dwyer and J. Nelson, "Roboflow," Roboflow, 2022. [Online]. Available: https://roboflow.com.

[36] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[37] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the International Conference on Machine Learning (ICML),* Sydney, NSW, Australia, 2017.

[38] H. G. Merkus, Particle Size Measurements, Dordrecht: Springer, 2010.

Appendix B: Hyperparameters

Hyperparameters of Mask R-CNN

```
MODEL:
  MASK_ON: True
  RESNETS:
    DEPTH: 101
SOLVER:
  STEPS: (36500, 38500)
  MAX_ITER: 40000
MODEL:
  META_ARCHITECTURE: "GeneralizedRCNN"
  BACKBONE:
    NAME: "build_resnet_fpn_backbone"
  RESNETS:
    OUT_FEATURES: ["res2", "res3", "res4", "res5"]
  FPN:
    IN_FEATURES: ["res2", "res3", "res4", "res5"]
  ANCHOR_GENERATOR:
    SIZES: [[32], [64], [128], [256], [512]]  # One size for each in feature map
    ASPECT_RATIOS: [[0.5, 1.0, 2.0]]  # Three aspect ratios (same for all in feature maps)
  RPN:
    IN_FEATURES: ["p2", "p3", "p4", "p5", "p6"]
    PRE_NMS_TOPK_TRAIN: 2000  # Per FPN level
    PRE_NMS_TOPK_TEST: 1000  # Per FPN level
    POST_NMS_TOPK_TRAIN: 1000
    POST_NMS_TOPK_TEST: 1000
  ROI_HEADS:
    NAME: "StandardROIHeads"
    IN_FEATURES: ["p2", "p3", "p4", "p5"]
  ROI_BOX_HEAD:
    NAME: "FastRCNNConvFCHead"
    NUM_FC: 2
    POOLER_RESOLUTION: 7
  ROI_MASK_HEAD:
    NAME: "MaskRCNNConvUpsampleHead"
    NUM_CONV: 4
    POOLER_RESOLUTION: 14
DATASETS:
  TRAIN: ("my_dataset_train",)
  TEST: ("my_dataset_val",)
SOLVER:
  IMS_PER_BATCH:4
  BASE_LR: 0.001
  STEPS: (60000, 80000)
  MAX_ITER: 90000
INPUT:
  MIN_SIZE_TRAIN: (640, 672, 704, 736, 768, 800)
VERSION: 2
```

Hyperparameters of Cascade Mask R-CNN

```
_BASE_: "../Base-RCNN-FPN.yaml"
MODEL:
  MASK_ON: True
  RESNETS:
    DEPTH: 50
  ROI_HEADS:
    NAME: CascadeROIHeads
  ROI_BOX_HEAD:
    CLS_AGNOSTIC_BBOX_REG: True
  RPN:
    POST_NMS_TOPK_TRAIN: 2000
SOLVER:
  STEPS: (36500, 38500)
  MAX_ITER: 40000
MODEL:
  META_ARCHITECTURE: "GeneralizedRCNN"
  BACKBONE:
    NAME: "build_resnet_fpn_backbone"
  RESNETS:
    OUT_FEATURES: ["res2", "res3", "res4", "res5"]
  FPN:
    IN_FEATURES: ["res2", "res3", "res4", "res5"]
  ANCHOR_GENERATOR:
    SIZES: [[32], [64], [128], [256], [512]]  # One size for each in feature map
    ASPECT_RATIOS: [[0.5, 1.0, 2.0]]  # Three aspect ratios (same for all in feature maps)
  RPN:
    IN_FEATURES: ["p2", "p3", "p4", "p5", "p6"]
    PRE_NMS_TOPK_TRAIN: 2000  # Per FPN level
    PRE_NMS_TOPK_TEST: 1000  # Per FPN level
    POST_NMS_TOPK_TRAIN: 1000
    POST_NMS_TOPK_TEST: 1000
  ROI_HEADS:
    NAME: "StandardROIHeads"
    IN_FEATURES: ["p2", "p3", "p4", "p5"]
  ROI_BOX_HEAD:
    NAME: "FastRCNNConvFCHead"
    NUM_FC: 2
    POOLER_RESOLUTION: 7
  ROI_MASK_HEAD:
    NAME: "MaskRCNNConvUpsampleHead"
    NUM_CONV: 4
    POOLER_RESOLUTION: 14
DATASETS:
  TRAIN: ("my_dataset_train",)
  TEST: ("my_dataset_val",)
SOLVER:
  IMS_PER_BATCH:4
  BASE_LR: 0.001
  STEPS: (60000, 80000)
  MAX_ITER: 90000
INPUT:
  MIN_SIZE_TRAIN: (640, 672, 704, 736, 768, 800)
VERSION: 2
```

Hyperparameters of ISTR

```
_BASE_: "Base-ISTR.yaml"
MODEL:
  RESNETS:
    DEPTH: 101
    STRIDE_IN_1X1: False
  ISTR:
    NUM_PROPOSALS: 300
    NUM_CLASSES: 4
DATASETS:
  TRAIN: ("my_dataset_train",)
  TEST:  ("my_dataset_val",)
SOLVER:
  STEPS: (36500, 38500)
  MAX_ITER: 40000
INPUT:
  FORMAT: "RGB"
CUDNN_BENCHMARK: False
DATALOADER:
  ASPECT_RATIO_GROUPING: True
  FILTER_EMPTY_ANNOTATIONS: True
  NUM_WORKERS: 4
  REPEAT_THRESHOLD: 0.0
  SAMPLER_TRAIN: TrainingSampler
DATASETS:
  PRECOMPUTED_PROPOSAL_TOPK_TEST: 1000
  PRECOMPUTED_PROPOSAL_TOPK_TRAIN: 2000
  PROPOSAL_FILES_TEST: ()
  PROPOSAL_FILES_TRAIN: ()
  TEST: ('my_dataset_val',)
  TRAIN: ('my_dataset_train',)
GLOBAL:
  HACK: 1.0
INPUT:
  CROP:
    ENABLED: True
    SIZE: [0.7, 0.7]
    TYPE: relative
  FORMAT: RGB
  MASK_FORMAT: polygon
  MAX_SIZE_TEST: 1333
  MAX_SIZE_TRAIN: 1333
  MIN_SIZE_TEST: 800
  MIN_SIZE_TRAIN: (480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800)
  MIN_SIZE_TRAIN_SAMPLING: choice
  RANDOM_FLIP: horizontal
MODEL:
  ANCHOR_GENERATOR:
    ANGLES: [[-90, 0, 90]]
    ASPECT_RATIOS: [[0.5, 1.0, 2.0]]
    NAME: DefaultAnchorGenerator
    OFFSET: 0.0
    SIZES: [[32, 64, 128, 256, 512]]
  BACKBONE:
    FREEZE_AT: 2
    NAME: build_resnet_fpn_backbone
  DEVICE: cuda
  FPN:
    FUSE_TYPE: sum
    IN_FEATURES: ['res2', 'res3', 'res4', 'res5']
    NORM:
    OUT_CHANNELS: 256
  ISTR:
    ACTIVATION: relu
    ALPHA: 0.25
    CLASS_WEIGHT: 2.0
    DEEP_SUPERVISION: True
    DIM_DYNAMIC: 64
    DIM_FEEDFORWARD: 2048
    DROPOUT: 0.0
    GAMMA: 2.0
    GIOU_WEIGHT: 2.0
    HIDDEN_DIM: 256
    IOU_LABELS: [0, 1]
    IOU_THRESHOLDS: [0.5]
    L1_WEIGHT: 5.0
    MASK_DIM: 60
    MASK_WEIGHT: 2.0
    NHEADS: 8
    NO_OBJECT_WEIGHT: 0.1
    NUM_CLASSES: 4
    NUM_CLS: 3
    NUM_DYNAMIC: 2
    NUM_HEADS: 6
    NUM_MASK: 3
    NUM_PROPOSALS: 300
    NUM_REG: 3
    PATH_COMPONENTS:
/content/ISTR/projects/ISTR/LME/coco_2017_train_class_agnosticTrue_whitenTrue_sigmoidTrue_60_
siz28.npz
    PRIOR_PROB: 0.01
  KEYPOINT_ON: False
  LOAD_PROPOSALS: False
  MASK_ON: True
  META_ARCHITECTURE: ISTR
  PANOPTIC_FPN:
    COMBINE:
      ENABLED: True
      INSTANCES_CONFIDENCE_THRESH: 0.5
      OVERLAP_THRESH: 0.5
      STUFF_AREA_LIMIT: 4096
    INSTANCE_LOSS_WEIGHT: 1.0
  PIXEL_MEAN: [123.675, 116.28, 103.53]
  PIXEL_STD: [58.395, 57.12, 57.375]
  PROPOSAL_GENERATOR:
    MIN_SIZE: 0
    NAME: RPN
  RESNETS:
    DEFORM_MODULATED: False
    DEFORM_NUM_GROUPS: 1
    DEFORM_ON_PER_STAGE: [False, False, False, False]
    DEPTH: 101
    NORM: FrozenBN
    NUM_GROUPS: 1
    OUT_FEATURES: ['res2', 'res3', 'res4', 'res5']
    RES2_OUT_CHANNELS: 256
    RES5_DILATION: 1
    STEM_OUT_CHANNELS: 64
    STRIDE_IN_1X1: False
    WIDTH_PER_GROUP: 64
  RETINANET:
    BBOX_REG_LOSS_TYPE: smooth_l1
    BBOX_REG_WEIGHTS: (1.0, 1.0, 1.0, 1.0)
    FOCAL_LOSS_ALPHA: 0.25
    FOCAL_LOSS_GAMMA: 2.0
    IN_FEATURES: ['p3', 'p4', 'p5', 'p6', 'p7']
    IOU_LABELS: [0, -1, 1]
    IOU_THRESHOLDS: [0.4, 0.5]
    NMS_THRESH_TEST: 0.5
    NORM:
    NUM_CLASSES: 80
    NUM_CONVS: 4
    PRIOR_PROB: 0.01
    SCORE_THRESH_TEST: 0.05
    SMOOTH_L1_LOSS_BETA: 0.1
    TOPK_CANDIDATES_TEST: 1000
  ROI_BOX_CASCADE_HEAD:
    BBOX_REG_WEIGHTS: ((10.0, 10.0, 5.0, 5.0), (20.0, 20.0, 10.0, 10.0), (30.0, 30.0, 15.0, 15.0))
    IOUS: (0.5, 0.6, 0.7)
  ROI_BOX_HEAD:
    BBOX_REG_LOSS_TYPE: smooth_l1
    BBOX_REG_LOSS_WEIGHT: 1.0
    BBOX_REG_WEIGHTS: (10.0, 10.0, 5.0, 5.0)
    CLS_AGNOSTIC_BBOX_REG: False
    CONV_DIM: 256
    FC_DIM: 1024
    NAME:
    NORM:
    NUM_CONV: 0
    NUM_FC: 0
    POOLER_RESOLUTION: 7
    POOLER_SAMPLING_RATIO: 2
    POOLER_TYPE: ROIAlignV2
    SMOOTH_L1_BETA: 0.0
    TRAIN_ON_PRED_BOXES: False
  ROI_HEADS:
    BATCH_SIZE_PER_IMAGE: 512
    IN_FEATURES: ['p2', 'p3', 'p4', 'p5']
    IOU_LABELS: [0, 1]
    IOU_THRESHOLDS: [0.5]
    NAME: Res5ROIHeads
    NMS_THRESH_TEST: 0.5
    NUM_CLASSES: 80
    POSITIVE_FRACTION: 0.25
    PROPOSAL_APPEND_GT: True
    SCORE_THRESH_TEST: 0.05
  ROI_KEYPOINT_HEAD:
    CONV_DIMS: (512, 512, 512, 512, 512, 512, 512, 512)
    LOSS_WEIGHT: 1.0
    MIN_KEYPOINTS_PER_IMAGE: 1
    NAME: KRCNNConvDeconvUpsampleHead
    NORMALIZE_LOSS_BY_VISIBLE_KEYPOINTS: True
    NUM_KEYPOINTS: 17
    POOLER_RESOLUTION: 14
    POOLER_SAMPLING_RATIO: 0
    POOLER_TYPE: ROIAlignV2
  ROI_MASK_HEAD:
    CLS_AGNOSTIC_MASK: False
    CONV_DIM: 256
    NAME: MaskRCNNConvUpsampleHead
    NORM:
    NUM_CONV: 0
    POOLER_RESOLUTION: 14
    POOLER_SAMPLING_RATIO: 0
    POOLER_TYPE: ROIAlignV2
  RPN:
    BATCH_SIZE_PER_IMAGE: 256
    BBOX_REG_LOSS_TYPE: smooth_l1
    BBOX_REG_LOSS_WEIGHT: 1.0
    BBOX_REG_WEIGHTS: (1.0, 1.0, 1.0, 1.0)
    BOUNDARY_THRESH: -1
    CONV_DIMS: [-1]
    HEAD_NAME: StandardRPNHead
    IN_FEATURES: ['res4']
    IOU_LABELS: [0, -1, 1]
    IOU_THRESHOLDS: [0.3, 0.7]
    LOSS_WEIGHT: 1.0
    NMS_THRESH: 0.7
    POSITIVE_FRACTION: 0.5
    POST_NMS_TOPK_TEST: 1000
    POST_NMS_TOPK_TRAIN: 2000
    PRE_NMS_TOPK_TEST: 6000
    PRE_NMS_TOPK_TRAIN: 12000
    SMOOTH_L1_BETA: 0.0
  SEM_SEG_HEAD:
    COMMON_STRIDE: 4
    CONVS_DIM: 128
    IGNORE_VALUE: 255
    IN_FEATURES: ['p2', 'p3', 'p4', 'p5']
    LOSS_WEIGHT: 1.0
    NAME: SemSegFPNHead
    NORM: GN
    NUM_CLASSES: 54
  WEIGHTS:
OUTPUT_DIR: ./output
SEED: 40244023
SOLVER:
  AMP:
    ENABLED: False
  BACKBONE_MULTIPLIER: 1.0
  BASE_LR: 5e-05
  BIAS_LR_FACTOR: 1.0
  CHECKPOINT_PERIOD: 5000
  CLIP_GRADIENTS:
    CLIP_TYPE: full_model
    CLIP_VALUE: 1.0
    ENABLED: True
    NORM_TYPE: 2.0
  GAMMA: 0.1
  IMS_PER_BATCH: 4
  LR_SCHEDULER_NAME: WarmupMultiStepLR
  MAX_ITER: 40000
  MOMENTUM: 0.9
  NESTEROV: False
  OPTIMIZER: ADAMW
  REFERENCE_WORLD_SIZE: 0
  STEPS: (36500, 38500)
  WARMUP_FACTOR: 0.01
  WARMUP_ITERS: 10
  WARMUP_METHOD: linear
  WEIGHT_DECAY: 0.0001
  WEIGHT_DECAY_BIAS: 0.0001
  WEIGHT_DECAY_NORM: 0.0
TEST:
  AUG:
    ENABLED: False
    FLIP: True
    MAX_SIZE: 4000
    MIN_SIZES: (400, 500, 600, 700, 800, 900, 1000, 1100, 1200)
  DETECTIONS_PER_IMAGE: 100
  EVAL_PERIOD: 500
  EXPECTED_RESULTS: []
  KEYPOINT_OKS_SIGMAS: []
  PRECISE_BN:
    ENABLED: False
    NUM_ITER: 200
VERSION: 2
VIS_PERIOD: 0
```