Edge Deep Learning Towards the Metallurgical Industry: Improving the Hybrid Pelletized Sinter (HPS) Process

Natália F. de C. Meira[1], Mateus C. Silva[1(✉)], Cláudio B. Vieira[1], Alinne Souza[2], and Ricardo A. R. Oliveira[1]

[1] Federal University of Ouro Preto, Ouro Preto, Brazil
mateuscoelho.ccom@gmail.com
[2] ArcelorMittal, João Monlevade, Brazil
alinne.souza@arcelormittal.com.br

Abstract. The implementation of intelligent systems in the processes brings the industries of the mining and metallurgy sectors closer to the context of Industry 4.0 and provides significant improvements, especially in the production and consumption of raw materials and internal products. In this work, we propose an Artificial Intelligence System in Deep Learning with Edge Computing to recognize the quasi-particles of the Hybrid Pelletized Sinter (HPS) process in the steel industry. We train our model with the aXeleRate tool using the Keras-Tensorflow framework and MobileNet architecture. We then tested the model in an embedded system using the SiPEED MaiX Dock board. The model validation results were 98.60% precision and 100.00% recall. Bench-scale test results were 100.00% precision and 70.00% recall. The results were promising and indicate the feasibility of the proposal.

Keywords: Artificial intelligence · AIoT · Edge computing

1 Introduction

The mining-metallurgical sector is one of the most traditional productive areas and in recent years, innovation and technology have developed new methods of production and development [2,12,19,24,27,28]. Thus, innovative projects are essential for the modernization of these processes, as they are of high economic interest. In the steel industry, one of the main process parameters is the particle size distribution of materials [35]. This concept means the size distribution of the present particles, which allows their employability in the productive process.

When transiting through the production plant, engineers and operators need to know the granulometric distribution continually. This information is essential as a process parameter or for making decisions under critical conditions. Along the steel industry process, the materials are transported using conveyor belts in many stages. These granulometric distribution changes can jeopardize the process if they are not within the required specifications [9].

J. Filipe et al. (Eds.): ICEIS 2021, LNBIP 455, pp. 149–167, 2022.
https://doi.org/10.1007/978-3-031-08965-7_8

Thus, the implementation of an algorithm in an embedded system that classifies quasi-particles according to their particle size distribution provides a way to solve this problem and improve the production process. Quasi-particles are micro agglomerates of materials formed in the HPS (Hybrid Pelletized Sinter) process [9,16]. We divided this problem into two steps: i) quickly identify the presence of a tray containing a sample of quasi-particles in the industrial sampler; ii) perform the particle size distribution of the material present in the tray. Therefore, the objective of this work is to propose the implementation of deep learning (DL) algorithm embedded in an edge computing device to classify images according to the presence or absence of quasi-particles samples in a tray, i.e., step i) of this process.

In this first conjecture, the user must photograph a sample of the material on the conveyor belt. The result is accessible through the display and also through a wireless network connection, which aims to classify that image as a quasi-particle sample with or without quasi-particle in the tray, or if there is another object causing interference. The use of artificial intelligence in edge computing devices is still an open problem, and the use of edge AI devices allows the expansion of deep learning to the IoT (Internet of Things) [5]. The implementation of an edge computing solution avoids a high throughput of data transmission. This trend takes the information and communication resources to the edge, with faster services and responses to the end-user [5].

The fast response to detected conditions enables a better process control. For instance, a granulometry pattern above the expected is an indicator of elevated moisture, which can cause clogging in the material transfer chutes between the conveyor belts. This event can paralyze the whole production process, exposing the operators to risk conditions and losing productivity.

In the industry's routine, this process can take a long time and does not guarantee quality. In many cases, this process takes substantial time changes, making it impossible to enable quick responses due to changes in production variables. In current applications, checking the particle size distribution of certain materials takes place through a manual process. In this task, an operator collects a sample of material from the production process and manually analyzes it with the aid of a series of sieves in a laboratory to obtain the particle size distribution. This procedure takes place several times a day, and the information obtained is used as a parameter for making decisions about the process.

Thus, manual analysis motivated the development of a DL-based device to detect the quasi-particle sample. We also incorporated this algorithm into a specialized edge computing device to detect quasi-particles from the Hybrid Pelletized Sınter (HPS) steelmaking process.

This work consists of the extended version of the paper [21] published in the ICEIS 2021 conference proceedings. Here, we organized the work to facilitate the reader's understanding of the methodological approach. As this work is an extended version, we analyzed further related works, creating a solid theoretical framework for our approach.

This paper is organized as follows: In Sect. 2, we review the literature and some ground concepts of this topic. Section 3 presents some of state-of the-art the related work. In Sect. 4, we present a description of the appliance features,

including the Deep Learning algorithm and the specialized hardware. In Sect. 5, we explain the employed experimental methodology. The results are presented in Sect. 6, and we present further discussions in Sect. 7.

2 Theoritical References

In this section, we present some theoretical references about the concepts applied to develop the proposed solution. This proposal's main element is a Convolutional Neural Network (CNN) applied to an edge computing solution. The proposal relates to the usage of an application in images of dense scenes. Thus, it is necessary to discuss both the issues related to the targeted problem itself and the matters related to the Edge AI concept.

Some of the problems faced in this matter are similar to others presented in the literature. For instance, we observed similar features from this work in precision agriculture appliances [11,25], and even in counting people in agglomeration [32]. Among the presented challenges, we enforce some aspects:

- Occlusion - often quasi-particles overlap, causing partial occlusion;
- Complex background - homogeneity in the shape, texture, or color of the background and objects;
- Rotation - images are often rotated at different angles;
- Lighting changes - images are exposed to different light levels during the day;
- Image resolution and noise - limits detection of small objects.

2.1 Deep Learning in Dense Scenes

Lecun et al. [14] state that Deep Learning (DL) is a set of techniques from the Machine Learning universe, often referred to as Artificial Intelligence. These algorithms' formalization comes from the Artificial Neural Networks (ANN), containing multiple hidden layers and massive training datasets. According to Zhang et al. [32], DL algorithms represent state of the art on Machine Learning techniques. Nonetheless, the detection of objects in dense scenes is particularly challenging.

Zhang et al. [32] separate dense scenes into two different classes: quantity dense scenes and internally dense scenes. In the first one, there is a large number of objects of interest in the scene. The second one happens when the objects have dense inner attributes. In both cases, labeling the data is a significant challenge, as the classification is affected by noise and resolution on small objects detection. According to these authors, the best DL architectures for classification in dense scenes are VGGNet, GoogLeNet, ResNet. Also, the best architecture for object detection are DetectNet and YOLO.

Gao et al. [7] analyzed 220 related works to understand the crowd counting process systematically. These authors point out that the main challenge is the detection of small objects in a scene. This trait happens as in crowd scenes, the individuals' heads are often too small. According to the authors, the most

successful techniques for counting crowds based on detection are SSD, YOLO, and RCNNs. Although these architectures had success in sparse scenes, these networks had unsatisfactory results given scenes with occlusion, disorder, and dense background. Furthermore, SSD is not efficient with small objects on the images, as its intermediate layers resource mapping may dilute the detected object's information. For the R-CNN, Zhou et al. [33] proposed an improvement based on PCA Jittering to enhance the detection of small objects on the Faster R-CNN architecture.

The presented work display some of the challenges in developing Convolutional Neural Networks (CNNs) capable of analyzing dense scenes with occluded objects. This issue is more significant when the dataset complexity increases. Developers often follow a synthetic database procedure to solve this problem, with further validation with actual real data. The obtained results are usually good, except if there is a substantial deviation from the synthetic and real datasets [32].

2.2 Edge AI Concepts

Another critical aspect of the solution is the algorithm persistence in edge computing applications. The evolution of embedded computing technologies raises the challenge of providing machine learning as services in edge applications with quality. Thus, the creation of reduced models and specialized hardware create the concept of an "Edge AI" [31]. This novel perspective targets using machine learning in edge devices with independence from cloud applications.

Nonetheless, developing machine learning and especially DL models for edge computing devices is a challenging task. Deep Neural Networks (DNNs) are generally computationally intensive models and require high computational power [15]. Moving this application to the cloud requires high data throughput through a network infrastructure. The growing number of devices can easily exceed network capabilities [17].

Zhou et al. [34] state that there are some issues to solve for enabling the Edge AI development. Among these challenges, we enforce:

- Programming and Software Platforms;
- Resource-Friendly Edge AI;
- Computational-Aware Techniques.

Another aspect to be considered when developing new edge computing solutions is hardware restrictions. As mentioned earlier, most DL architectures require high computational performance. One result of this problem is the integration of dedicated hardware to optimize Edge AI solutions [4,10,20,22].

The approach and recent availability of Edge AI solutions have contributed to reconciling the concepts of edge computing and AI, and allow critical computing and latency AI-based applications to run in real-time [1,30]. The authors consider that edge and cloud are complementary: in the division of the AI lifecycle workflow, we can deploy model training in the cloud and perform inference at the edge.

Cornetta and Touhafi [4] presented a review of the most popular machine learning algorithms to run on resource-constrained embedded devices. The deep learning techniques used in IoT devices were Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) and, for the authors, solutions based on TensorflowLite are not yet fully implementable in embedded devices.

The work by Liu et al. [18] proposed a multisensor data anomaly detection method based on edge computing in underground mining. In general, IoT technology is widely used in underground mining construction safety monitoring and early warning, however, some problems are associated with data anomalies, such as i) sensor failures, ii) environmental changes, and; iii) wireless data interference. Other problems are associated with cloud processing: i) amount of invalid and redundant data transmission that wastes limited network resources, ii) some sensor data that has real-time requirements for detecting anomalies that may be delayed, iii) latency to the cloud can be prohibitive for delay-sensitive applications and, iv) the transfer of sensitive data retrieved by IoT devices can raise privacy issues [1,18,30].

Lin et al. [17] implemented a YOLOv3-based pavement defect detection system in an embedded Xilinx ZCU104 system. The authors compacted the model without significantly reducing accuracy by the quantization method, reducing the size of the original model by 23% and comparing performance on the Xilinx ZCU104 with an embedded Nvidia TX2 system. The running speed of Xilinx ZCU104 was 27.4 FPS, which met the requirements of low power consumption and real-time response.

Cob-Parro et al. [3] presented an intelligent video surveillance system to detect, count, and track people in real-time on an embedded hardware system with vision processing units (VPUs) modules on the UpSquared2 embedded platform and MobileNet-SSD architecture for the task. The model achieved an mAP (Mean Average Precision) of 72.7%. Edge AI performance on CPU was 13.93 ms while on VPU it was 8.71 ms.

3 Related Work

Given the importance of the iron ore agglomeration stage for the later stages of the process, several studies have been carried out to control and monitor the variables that interfere in the sintering and pelletizing processes.

Dias [6] proposed a granulometric control system for iron ore pellets by controlling the water injection in the pellet drum, which, until then, was done manually by the operators according to the need of the process. The results showed that water addition tends to increase the pellets' granulometry and that the control tends to homogenize the pellets. However, for the controlled variable to present stabilization, it would be necessary to study other parameters, such as water saturation due to pellet recirculation outside the required particle size range.

Studies on the influence of raw materials in the cold agglomeration process of the HPS process were also studied, as shown in Januzzi [9]. The work had the objective to characterize the raw materials, study the contribution of

each of them in the cold agglomeration process, and adjust the parameters to improve the process's performance. One of the measures taken was the changes in the granulometric distribution curves of serpentinite, limestone, and manganese ore, which promoted an improvement in the quasi-particles' average size. Consequently, this measure causes "a positive effect on the suction pressure in the sinter allowing the increase of layer height, gain in productivity and sinter production" [9], once again demonstrating the importance of granulometric distribution in the iron ore agglomeration process.

For the case where the manual control depended on the area operators to obtain the adequate granulometry of the raw pellet, Passos et al. [23] developed its work in the implementation of an advanced control system (SCAP) intending to control the granulometry of the pellets raw materials acting on the speed and feeding of the disks. The results showed the stability of the production process, mainly in controlling the pellets' granulometric distribution, the stability of the dosage of inputs, and the hardening furnace's increased permeability.

Souza [29] proposed the use of deep learning algorithms to identify iron ore particles and measure their linear dimensions from images obtained in the primary crushing operation. The authors evaluated the SSD, Faster R-CNN, YOLOv3, and U-NET algorithms. The particles from the bench images consisted of 4.8 mm to 19 mm fragments and the fragments from the industrial area video images had dimensions greater than 200 mm. The results obtained in the training of SSD, Faster R-CNN, and YOLOv3 networks showed low accuracy and low assertiveness index. The U-NET network had an accuracy of 91.3%. From the generated masks, the authors developed a routine with the OpenCV computer vision library to generate a bounding box over the mask and supply the side length of the box to measure the object.

Other works aimed to obtain the particle size distribution by images in iron ore agglomeration processes. For example, to characterize ultra-fine materials and medium-sized consumption, Gontijo [8] performed prior image processing in a Scanning Electron Microscope (SEM). The image particles were digitized, scaled in software, classified by color into size ranges (intervals), and, after classification, generated graphs of particle size distributions.

The work by Santos et al. [26] proposed an automatic image analysis routine to identify the sintering quasi-particles and classify them into three classes, calculate the fraction of the class area, circularity, and thickness of the adherent layer, and, finally, quantify the mineral phases present in the quasi-particle nuclei. The authors used samples produced in a pilot sinter plant, which were classified into the following size ranges: >4.76 mm, 2.83–4.76 mm, and 1.00–2.83 mm, and the size fraction of >1.0 mm was discarded.

Images were acquired by light reflected light microscopy with approximately 50x magnification and resolution of 2.05 μm/pixel. For digital image processing and analysis, the authors used the Fiji image processing package. With the computer used, the developed routine was able to process a 4.76 mm grain image in about 6 min, while a 1.00 mm grain image took about 18 min due to the increased number of particles.

In the final result, the authors considered that the developed routine provided good performance and speed, compared to human performance, as the system was able to process 1.00 mm samples in about 20 min, while an operator can take up to 6 h. Santos et al. (2019) concluded the work considering a future work with the use of Convolutional Neural Networks (CNN) for segmentation, as "CNNs can achieve high efficiency in classification and segmentation problems, combining and sometimes exceeding performance human, as they are capable of processing highly abstract resources" [26].

4 Edge AI Hardware

In this work, we decided to implement the solution using the SiPEED MAiX Dock board, displayed in Fig. 1. Some performance numbers of the board are shown in Table 1. The work of Klippel et al. [13] demonstrates the comparison between SiPEED MaiX BiT, Raspberry Pi 3, and Jetson Nvidia Nano cards. The authors implemented the SiPEED MaiX BiT for the detection of tears in conveyor belts. The SiPEED MaiX Dock board is similar to the one used in this work, and we follow the methodology proposed by Klippel et al. [13].

Fig. 1. SiPEED M1 Dock - demonstration [21].

Table 1. Embedded platform performance numbers [21].

Parameter	Characteristics
CPU	64-bit RISC-v processor and core
Chipset	K210 - RISC - V
Image recognition	qvg at 60 fps/vg at 30 fps
Clock (GHz)	0.40
AI resources	KPU
OS/Language	uPython
Dimensions (mm)	60 × 43 × 5

This platform has an onboard device with artificial intelligence (AI) hardware acceleration. MAiX is the module explicitly developed for SiPEED, designed to perform AI. It offers high performance considering a small physical and energy area, allowing the implantation of high precision AI and a competitive price. The main advantages of this device are:

- Complete hardware and software infrastructure to facilitate the deployment of AI-based solutions;
- Good performance, small size, low energy consumption, and low cost, which allows a broad deployment of high quality AI on board;
- It can be used for an increasing number of industrial use cases, such as predictive maintenance, anomaly detection, machine vision, robotics, and voice recognition.

The SiPEED MAiX acts as the master controller, and the hardware has a KPU K210. MaixPy is a framework designed for AIoT programming, prepare on an AIoT K210 chip, and based on the Micropython syntax. MicroPython is a lean and efficient implementation of the Python 3 programming language, which includes a small subset of the standard Python library, and is optimized to run on microcontrollers and in restricted environments, facilitating programming on the K210 hardware. MAiX supports a fixed-point model that a conventional training structure trains according to specific restriction rules and has a model compiler to compile models in its model format. It is compatible with network architectures Tiny-Yolo and MobileNet-v1.

The Kendryte K210 is a dual-core RISCV64 SoC with AI capability that has machine vision capabilities and can perform low energy consumption Convolutional Neural Networks (CNNs) calculations, with features for object detection, image classification, detection and face recognition, obtaining target size and coordinates in real-time and obtaining the type of target detected in real-time. The KPU is a generalpurpose neural network processor with internal convolution, normalization, activation, and pooling operations. According to the manufacturer, it also has the following characteristics:

- Supports the fixed-point model that the conventional training structure trains according to specific restriction rules;
- There is no direct limit on the number of network layers, and each layer of the convolutional neural network parameters can be configured separately, including the number of input and output channels, the width of the input and output line, and the height of the column;
- Support for 1×1 and 3×3 convolution kernels;
- Support for any form of activation function;
- The maximum size of the supported neural network parameter for real-time work is from 5 MiB to 5.9 MiB.

This work's main contribution is the implementation of a deep learning method on an edge device for application aimed at the industrial environment, including practical tests on embedded hardware.

5 Experimental Metodology

This section assesses the experimental methodology used to validate the appliance, given the targeted hardware. For this matter, we present the employed

dataset, training process, and evaluation metrics. We test a pilot application classifier's performance and validate the model's transfer into the desired hardware.

5.1 Dataset

We did not find any available database of iron ore quasi-particles or micro-agglomerates. Therefore, one of our contributions was establishing a method to elaborate a dataset with real images of an industrial environment. The images used in the classifier training were elaborated from quasi-base reals in the industrial environment, and synthetic images were created on a bench scale. In the production process, a sampler removed several of the quasi-particles in trays with the help of an operator. These samples are taken to a nearby environment and photographed following a pre-established pattern.

We generated a dataset with 1368 images to create a pilot appliance, containing 1140 for training and 228 for validation (80/20 ratio). The dataset has three different classes: quasi-particle, non-category, and empty. We also added 343 synthetic images produced on the benchscale for the quasi-particle class training, as presented in Fig. 2. These images were generated to avoid the problems of overlapping and occlusion of the particles. We also added another 343 images of samples of quasi-particles carried out in a company in the mining-metallurgical sector with real data to contribute to the quasi-particle training dataset.

(a) (b)

Fig. 2. Images of quasi-particles trays (main class), in: a) real industrial image; b) synthetic image produced on a bench scale.

5.2 Training the Deep Learning Model

We conducted the training of the deep learning model on the Google Collaboratory platform. This process was carried out using the aXeleRate[1] tool. This application is a tool for training classification and detection models developed using the Keras/Tensorflow framework.

To perform the desired task, we chose to use the MobileNet as CNN architecture. We used version 0.75 MobileNet-224 v1, configured as a classifier, with 224 inputs, two layers fully connected with 100 and 50 neurons, and a dropout of 0.5. The training session held thirty epochs, and the learning rate adopted was 0.001. The initial weights of the model were loaded, considering the previous training with the ImageNet dataset. Also, data augmentation was performed during the training.

5.3 Edge AI Construction

For training, we implemented the aXeleRaTe framework, a Keras based framework for AI on the Edge, to run computer vision applications (image classification, object detection, semantic segmentation) on edge devices with hardware acceleration. AXeleRate simplifies the training and conversion of computer vision models and is optimized for workflow on the local machine and Google Colab. Supports conversion of trained model to: *.kmodel* (K210) and *.tflite* formats.

Figure 3 displays the process of using aXeleRate, with the main steps indicated by the blue circles. In (1), the dataset is loaded from Google Drive for training in the Keras-Tensorflow framework. Then (2), the model is delivered in

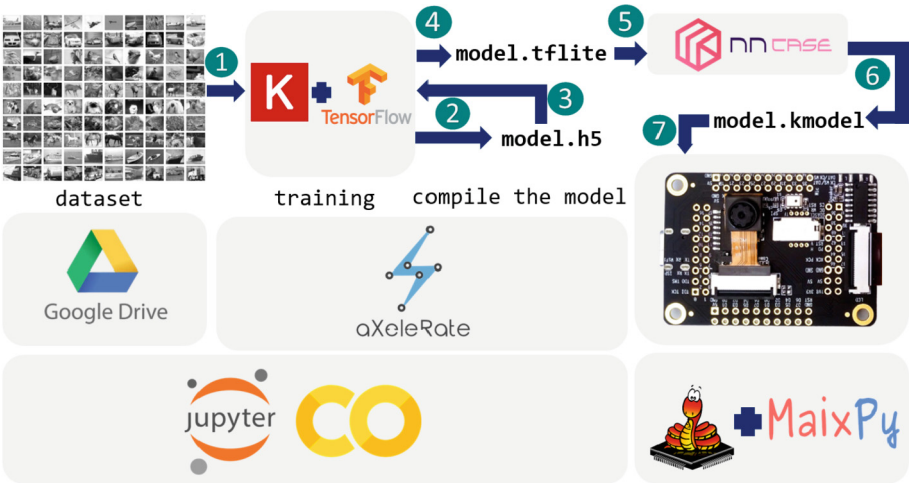

Fig. 3. Training and compilation with aXeleRate [21].

[1] https://github.com/AIWintermuteAI/aXeleRate.

the *.h5* format for classification and returns to Tensorflow (3) to be converted into the *.tflite* format (4). Thus, it is delivered to nncase (5) to be compiled into the format *.kmodel* (6), which is executed by KPU (7).

We assembled a SiPEED Dock plate for the execution of the bench-scale model with synthetic images. For this test, we used two Python scripts used for the tests. The first to capture photos with 224×224 resolution and storage on the SD card. The second to test the model from the storage data set previously stored on the SD card.

5.4 Evaluation Metrics

At first, the classification model's performance was calculated using the Confusion Matrix, which shows the classification frequencies for each class of the model. From this data, we extract the parameters: *precision*, given by 1, *recall* given by 2 and *F1*, given by 3. These parameters define how well the model worked, how good the model is for predicting positives, and the balance between the *precision* and the *recall* of the model.

For this matter, we followed the presented definitions: TP is a true-positive sample, FP is a false-positive sample, TN is a true-negative sample, and FN is a false-negative. TP occurs when the main class prediction is correct, and FP when it is mispredicted. TN occurs when the alternative class prediction is correct and FN when it is mispredicted.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 * \frac{precision \ * \ recall}{precision \ + \ recall} \tag{3}$$

6 Results

We present here the obtained results from the application of this procedure. Our preliminary results indicate the system feasibility and show the constraints to transport the model into the Edge AI device.

6.1 Training Model Performance

The training elapsed time was 54 min, reaching an accuracy of 98.60%. Figure 4 displays the evolution of the accuracy throughout the training stage. As displayed in the graph, the model's training converged in just ten iterations, indicating that the model had no great difficulty in differentiating the classes of images present in the database.

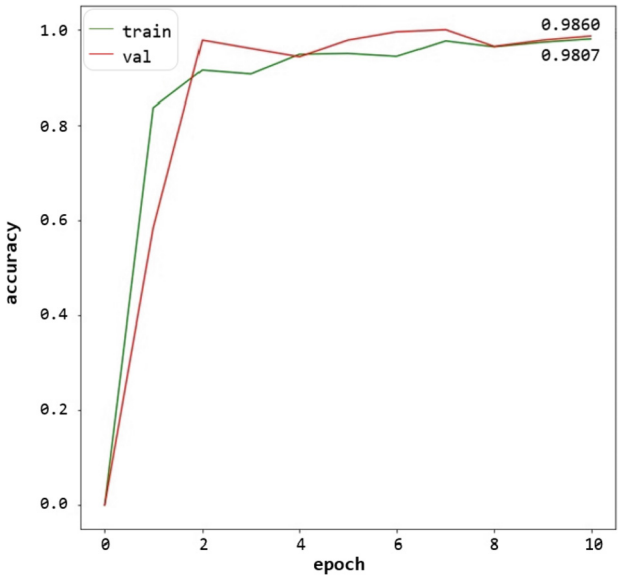

Fig. 4. Metrics for the training process [21].

To validate the model, we created a dataset with 228 images (Table 2). These frames were divided into three classes, containing 76 images each class: quasi-particle, non-category, and empty (empty refers to the same tray, but without the presence of quasiparticles). Table 3 displays the confusion matrix considering quasi-particles as the main class and Table 4 shows the performance indicators.

Table 2. Distribution of images in the dataset by class.

Dataset	Class	Number of images per class	Source	Total number of images
Validation set	quase_particle	76	38 real 38 synthetic	228
	non_category	76		
	Empty	76		

Table 3. Confusion matrix of model - validation set [21].

	Predict			
		quase_particle	non_categpry	Empty
Real	quase_particle	76	0	0
	non_category	1	74	1
	Empty	0	0	76

Table 4. Trained model performance at validation set [21].

Indicator	Value
Precision	98,60%
Recall	100%
F1	99,34%

The model precision was 98.60%. The application displayed problems in classifying some uncategorized images with quasi-particle and empty trays. The data suggest a good recall, which means that the model had a small error rate in the quasi-particles' classification when they were indeed quasi-particles. These results demonstrate the feasibility of the recognition process using the proposed dataset. This value enabled a balance in the F1 score.

6.2 Model Performance at Edge AI

We also tested the performance of the classifier in the edge computing candidate platform. After training, we loaded the model into the SiPEED Maix Dock for testing, as showed in Fig. 5. For this matter, we tested the system using images from the three classes (quasi-particle, non-category, and empty). Table 5 displays the confusion matrix and Table 6 shows the performance indicators.

In contrast to the value achieved in the validation set, or recall in the test set dropped to 70%, evaluated from the SiPEED embedded system. This result indicates that the model had to test positively for image simulations similar to industrial environment images, as specified in Figs. 6 and 7, although for synthetic images with spaced particles there was no difficulty, as defined in Fig. 8.

The work of Klippel et al. [13] implemented the SiPEED MaiX BiT to detect failures in conveyor belts. Our results for training performance are similar to the results obtained by Klippel et al. In the test performance, we obtained a lower recall, as shown.

The recall value in the tests does not match the results obtained in the tests carried out by Klippel et al. [13] To justify the value of 70%, we understand that the data set can be improved to only real images in future analyses. Also, there is a possibility of overfitting during training. In order to verify this hypothesis, we intend to increase the database in future works.

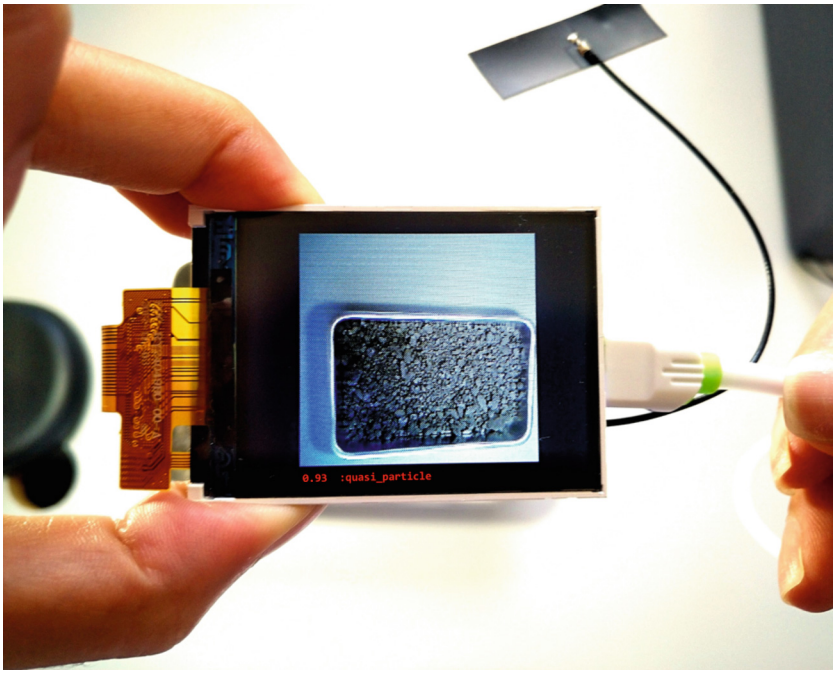

Fig. 5. SiPEED MaiX Dock - test demonstration [21].

Table 5. Confusion matrix of model - test set [21].

	Predict			
		quase_particle	non_categpry	Empty
Real	quase_particle	7	2	1
	non_category	0	9	1
	Empty	0	0	10

Table 6. Trained model performance at test set [21].

Indicator	Value
Precision	100,00%
Recall	70,00%
F1	82,35%

These data demonstrate the difficulty of reconciling results obtained on a bench scale with results close to real environments.

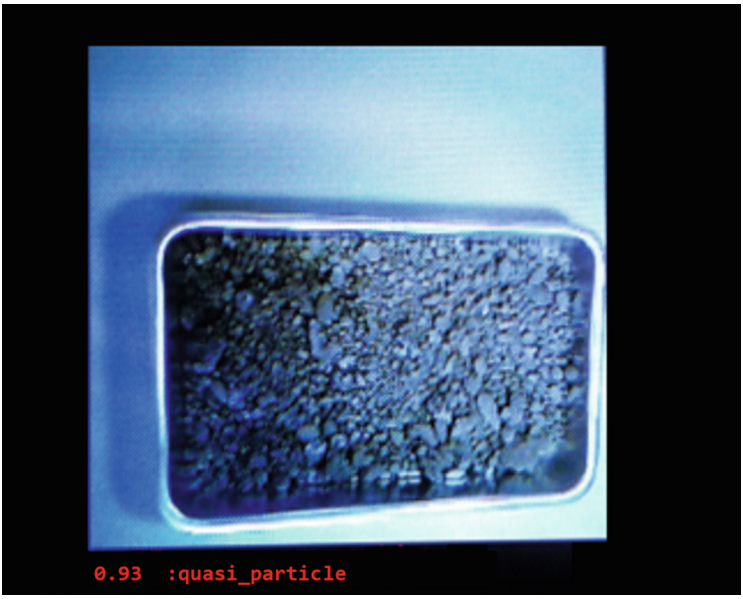

Fig. 6. Example of recognition of quasi-particles simulating sampling in an industrial environment during the test using SiPEED [21].

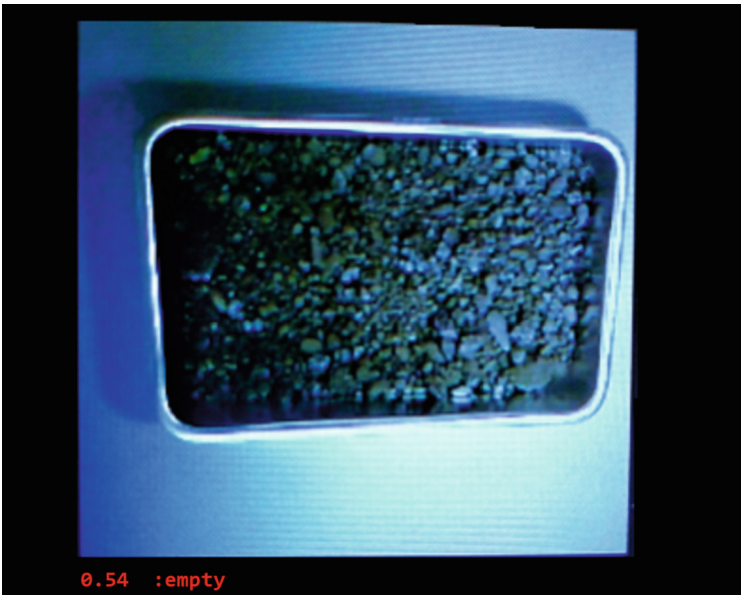

Fig. 7. Example of error in recognizing quasi-particles simulating sampling in an industrial environment during the test using SiPEED [21].

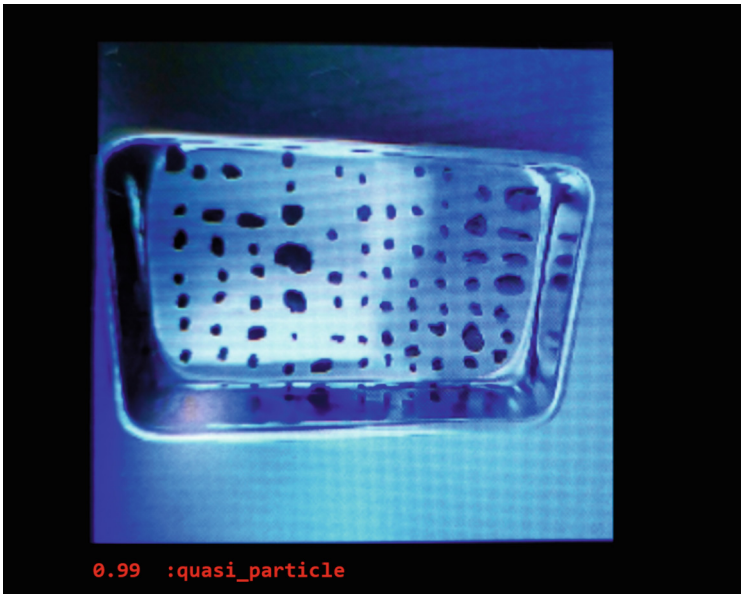

Fig. 8. Example of recognition of quasi-particles with sample developed on a bench scale during the test using SiPEED [21].

7 Conclusion

In this work, we implement the first stage of the pipeline to perform the recognition of quasi-particle images, with the identification of the sample through images with Deep Learning (DL). The objective was to classify trays containing quasi-particles, allowing them to differentiate themselves from other objects and even from empty trays. We train our model for and embed it to perform real-time inference on specialized Edge AI hardware. The advantages are (i) the start of the pipeline for automatic detection of industrial samples that are taken for particle size analysis, as this activity is performed manually; (ii) Edge AI embedded hardware implementation; (iii) solution developed for real-time inferential.

In developing the solution, we implemented a Convolutional Neural Network (CNN) to classify the images obtained in the industry and in a bench-scale to classify three situations. The main class is the recognition of the sample containing the process quasi-particles. The trained, validated, and evaluated model was embedded in an Edge Computing device for testing and evaluated again. The dataset images comprise situations such as dense scenes, problems such as occlusion, complex background changes, and light variations. Although there are wide applications of DL in dense scenes, there are still open questions to be resolved in the research process.

Deep learning models are computationally intensive. To perform real-time edge inference, we tested our application on the embedded SiPEED MaiX Dock board. This board features hardware and software infrastructure to enhance

Edge AI application development. Tests with SiPEED allow the detection of quasi-particles in synthetic images without difficulties, with the spaced distribution of particles and control of variability in the environment, such as luminosity. However, tests with real images had some flaws, evidenced by the drop in recall to 70%. Overfitting may have occurred during or training or, during tests, influences associated mainly with daylight, occlusion between particles, color homogeneity, and overlapping between objects.

Our work contributed to the implementation and evaluation of a work developed with a dataset of real images of the steel industry. Collecting data in an industrial environment can be challenging, and in the early stages of development, researchers sometimes choose to obtain their synthetic data in a bench-scale and controlled environment. From the results obtained in this step, it was possible to raise new hypotheses of approaches to improve the deep learning algorithm. Furthermore, the results were promising and indicate the feasibility of the proposal. We are in development for future work on the segmentation of quasi-particles in the samples by size classes.

Acknowledgements. The authors would like to thank CAPES, CNPq and the Federal University of Ouro Preto for supporting this work. Also, the authors would like to thank ArcelorMittal Monlevade for enabling the creation of a dataset with real images. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Finance code 308219/2020-1.

References

1. Campolo, C., Genovese, G., Iera, A., Molinaro, A.: Virtualizing AI at the distributed edge towards intelligent IoT applications. J. Sens. Actuator Netw. **10**(1), 13 (2021)
2. Chen, Z., Liu, L., Qi, X., Geng, J.: Digital mining technology-based teaching mode for mining engineering. iJET **11**(10), 47–52 (2016)
3. Cob-Parro, A.C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., Bravo-Muñoz, I.: Smart video surveillance system based on edge computing. Sensors **21**(9), 2958 (2021)
4. Cornetta, G., Touhafi, A.: Design and evaluation of a new machine learning framework for IoT and embedded devices. Electronics **10**(5), 600 (2021)
5. Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., Zomaya, A.Y.: Edge intelligence: the confluence of edge computing and artificial intelligence. IEEE Internet Things J. **7**, 7457–7469 (2020)
6. Dias, Í.D.S.M.: Sistema de controle granulométrico de pelotas de minério de ferro (2018)
7. Gao, G., Gao, J., Liu, Q., Wang, Q., Wang, Y.: CNN-based density estimation and crowd counting: a survey. arXiv preprint arXiv:2003.12783 (2020)
8. Gontijo, M.D.: Análise granulométrica por imagem de amostras ultrafinas. Rev. Engenharia Interesse Soc. **1**(3) (2018)
9. Januzzi, A.: Análise da aglomeração a frio no processo hps (hybrid pelletized sinter) com ênfase nas matérias-primas envolvidas (2008)

10. Karras, K., et al.: A hardware acceleration platform for AI-based inference at the edge. Circ. Syst. Signal Process. **39**(2), 1059–1070 (2020)
11. Keresztes, B., Abdelghafour, F., Randriamanga, D., da Costa, J.P., Germain, C.: Real-time fruit detection using deep neural networks. In: 14th International Conference on Precision Agriculture (2018)
12. Kinnunen, P.H.M., Kaksonen, A.H.: Towards circular economy in mining: opportunities and bottlenecks for tailings valorization. J. Clean. Prod. **228**, 153–160 (2019)
13. Klippel, E., Oliveira, R., Maslov, D., Bianchi, A., Silva, S.E., Garrocho, C.: Towards to an embedded edge AI implementation for longitudinal rip detection in conveyor belt. In: Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais, pp. 97–102. SBC, Porto Alegre (2020). https://doi.org/10.5753/sbesc_estendido.2020.13096, https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13096
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
15. Li, E., Zeng, L., Zhou, Z., Chen, X.: Edge AI: on-demand accelerating deep neural network inference via edge computing. IEEE Trans. Wirel. Commun. **19**(1), 447–457 (2019)
16. Lima, A.J.D.A.S.: Caracterização tecnológica de uma mistura de sinter feed e pellet feed para uso em processo hps (2019)
17. Lin, X., Li, J., Wu, J., Liang, H., Yang, W.: Making knowledge tradable in edge-AI enabled IoT: a consortium blockchain-based efficient and incentive approach. IEEE Trans. Industr. Inf. **15**(12), 6367–6378 (2019)
18. Liu, C., Su, X., Li, C.: Edge computing for data anomaly detection of multi-sensors in underground mining. Electronics **10**(3), 302 (2021)
19. Mardonova, M., Choi, Y.: Review of wearable device technology and its applications to the mining industry. Energies **11**(3), 547 (2018)
20. Mazzia, V., Khaliq, A., Salvetti, F., Chiaberge, M.: Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application. IEEE Access **8**, 9102–9114 (2020)
21. Meira, N., Silva, M., Oliveira, R., Souza, A., D'Angelo, T., Vieira, C.: Edge deep learning applied to granulometric analysis on quasi-particles from the hybrid pelletized sinter (HPS) process. In: Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS, pp. 527–535. INSTICC, SciTePress (2021). https://doi.org/10.5220/0010458805270535
22. Ohbuchi, E.: Low power AI hardware platform for deep learning in edge computing. In: 2018 IEEE CPMT Symposium Japan (ICSJ), pp. 89–90. IEEE (2018)
23. Passos, L.A.S., Moreira, J.L., Jorge, A., Cavalcante, M.V.S.: Melhoria no desempenho do processo de produção de pelotas de minério de ferro em discos de pelotização pela utilização de sistemas otimizantes com lógica nebulosa. In: ABM Proceedings. Editora Blucher, September 2014. https://doi.org/10.5151/2594-357x-25340
24. Robben, C., Wotruba, H.: Sensor-based ore sorting technology in mining-past, present and future. Minerals **9**(9), 523 (2019)
25. Saleem, M.H., Potgieter, J., Arif, K.M.: Plant disease detection and classification by deep learning. Plants **8**(11), 468 (2019)
26. Santos, R.B.M., Augusto, K.S., Paciornik, S., Alcantara Domingues, A.L.: An image analysis system for automatic characterisation of iron ore sintering quasi-particles. Mineral Process. Extract. Metallurgy 1–9 (2019)

27. Shibuta, Y., Ohno, M., Takaki, T.: Computational metallurgy: advent of cross-scale modeling: high-performance computing of solidification and grain growth. Adv. Theory Simul. **1**(9), 1870020 (2018)
28. Sinoviev, V.V., Okolnishnikov, V.V., Starodubov, A.N., Dorofeev, M.U.: Approach to effectiveness evaluation of robotics technology in mining using discrete event simulation. Int. J. Math. Comput. Simul. **10**, 123–128 (2016)
29. Souza, L.E.R.: Medição de granulometria de minério de ferro através de imagens em circuito de britagem primária (2020)
30. Wang, X., Han, Y., Leung, V.C., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tutor. **22**(2), 869–904 (2020)
31. Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., Chen, M.: In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Netw. **33**(5), 156–165 (2019)
32. Zhang, Q., Liu, Y., Gong, C., Chen, Y., Yu, H.: Applications of deep learning for dense scenes analysis in agriculture: a review. Sensors **20**(5), 1520 (2020)
33. Zhou, X., Fang, B., Qian, J., Xie, G., Deng, B., Qian, J.: Data driven faster R-CNN for transmission line object detection. In: Ning, H. (ed.) CyberDI/CyberLife -2019. CCIS, vol. 1137, pp. 379–389. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-1922-2_27
34. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., Zhang, J.: Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc. IEEE **107**(8), 1738–1762 (2019)
35. Zobnin, N.N., Torgovets, A.K., Pikalova, I.A., Yussupova, Y.S., Atakishiyev, S.A.: Influence of thermal stability of quartz and the particle size distribution of burden materials on the process of electrothermal smelting of metallurgical silicon. Orient. J. Chem. **34**(2), 1120–1125 (2018)