Edge Deep Learning Applied to Granulometric Analysis on Quasi-particles from the Hybrid Pelletized Sinter (HPS) Process

Natália F. de C. Meira[1][a], Mateus C. Silva[2][b], Ricardo A. R. Oliveira[2][c], Aline Souza[3], Thiago D'Angelo[2] and Cláudio B. Vieira[1]

[1]*Metallurgical Engineering Department, Federal University of Ouro Preto, Ouro Preto, Brazil*
[2]*Department of Computer Science, Federal University of Ouro Preto, Ouro Preto, Brazil*
[3]*ArcelorMittal, João Monlevade, Brazil*

Keywords: AIoT, Artificial Intelligence, Edge Computing, Edge Learning, Computer Vision.

Abstract: The mining and metallurgical industry seeks to adapt to Industry 4.0 with the implementation of Artificial Intelligence in the processes. The purpose of this paper is to develop the first steps of an Artificial Intelligence in Deep Learning with Edge Computing to recognize the quasi-particles from the Hybrid Pelletized Sinter (HPS) process and provide its particle size distribution. We trained our model with the aXeleRate tool using the Keras-Tensorflow framework and the MobileNet architecture and tested it with an embedded system using the SiPEED MaiX Dock board. Our model obtained 98.60% accuracy in training validation using real and synthetic images and 100% accuracy in tests with synthetic images and 70% recall. The tests' results indicate the feasibility of the proposed system, but with probable overfitting in the training stage.

1 INTRODUCTION

The mining and metallurgical sectors are some of the most traditional productive areas (Kinnunen and Kaksonen, 2019). In the later years, innovation and technology have developed new production and development methods (Robben and Wotruba, 2019; Mardonova and Choi, 2018; Sinoviev et al., 2016; Chen et al., 2016; Shibuta et al., 2018). Thus, innovative projects are crucial for these processes, as they have a high economic interest. In the steel industry, one of the primary process parameters is the granulometric distribution of materials (Zobnin et al., 2018). 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. By itself, the process provokes many variations in the materials' features due to humidity, source of the material, among other factors. These granulometric distribution changes can jeopardize the process if they are not within the required specifications (Januzzi, 2008).

Thus, implementing an algorithm in an embedded system that classifies the grains according to their granulometric distribution provides a path to solve this problem and improve the production process. For this matter, we propose using a deep learning algorithm embedded in an optimized computing device to classify image samples according to their granulometric distribution. The usage of artificial intelligence on edge computing devices is still an open problem, and the usage of specialized Edge AI devices allows the expansion of deep learning towards the IoT (Internet of Things) (Deng et al., 2020).

In this first conjecture, the user must photograph a sample of the material in the conveyor belt. The result is accessible through display and also through a wireless network connection. The implementation of an Edge Computing solution avoids a large data transmission throughput. This trend pushes the computing and communication resources to the edge, with faster

[a] https://orcid.org/0000-0002-7331-6263
[b] https://orcid.org/0000-0003-3717-1906
[c] https://orcid.org/0000-0001-5167-1523

Meira, N., Silva, M., Oliveira, R., Souza, A., D'Angelo, T. and Vieira, C.
Edge Deep Learning Applied to Granulometric Analysis on Quasi-particles from the Hybrid Pelletized Sinter (HPS) Process.
DOI: 10.5220/0010458805270535
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 527-535
ISBN: 978-989-758-509-8

services and answers to the final user (Deng et al., 2020).

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 significant amount of time and a lack of quality guarantee. In many cases, this process takes substantial time shifts, making it impossible to enable quick responses due to production variable changes. In the current applications, the verification of certain materials' granulometric distribution happens through a manual process. Figure 1 displays a real sample of the material obtained from the productive process.

Figure 1: Real Sample example taken from one of the stages of the steel industry process.

In this task, one operator sample material from the productive process and manually analyze it in the laboratory to obtain the granulometric distribution. This procedure happens several times during the day, and the obtained information is used as a parameter for the decision making process.

Thus, the manual analysis motivated the development of a Deep-Learning-based appliance to detect quasi-particles. Quasi-particles are material agglomerates formed in the HPS process (Januzzi, 2008). We also embedded this algorithm on a specialized edge computing device to detect quasi-particles from the Hybrid Pelletized Sínter (HPS) process from the steel industry.

This work is relevant because it consists of implementing a deep learning method in an edge device for application aimed at the industrial environment, including practical tests on embedded hardware.

This paper is organized as follows: In Section 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 Section 4, we present a description of the appliance features, including the Deep Learning algorithm and the specialized hardware. In Section 5, we explain the employed experimental methodology. The results are presented in Section 6, and we present further discussions in Section 7.

2 THEORETICAL 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.

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 (Keresztes et al., 2018; Saleem et al., 2019), and even in counting people in agglomeration (Zhang et al., 2020). Among the presented challenges, we enforce some aspects:

- Occlusion: Often, quasi-particles overlap one another, causing partial occlusion;
- Complex Background: Homogeneity in shape, texture, or color from the background and the objects;
- Rotation: Images are ofter rotated in different angles;
- Illumination Changes: Images are exposed to different levels of light throughout the daytime.

2.1 Deep Learning in Dense Scenes

Lecun et al. (LeCun et al., 2015) 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. (Zhang et al., 2020), 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. (Zhang et al., 2020) 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. (Gao et al., 2020) 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 R-CNNs. 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. (Zhou et al., 2019a) 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 (Zhang et al., 2020).

2.2 Edge AI

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" (Wang et al., 2019). 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) usually are computationally intensive models (Li et al., 2019). On the one hand, DNNs usually require much computational power. On the other hand, moving this application to the cloud requires a high data throughput through a network infrastructure. The increasing number of devices can easily exceed the networking capacities (Lin et al., 2019).

Zhou et al. (Zhou et al., 2019b) 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 in the development of novel edge computing solutions is the hardware constraints. As stated before, most DL architectures require a high computational charge. An outcome for this problem is integrating dedicated hardware to optimize Edge AI solutions (Mazzia et al., 2020; Ohbuchi, 2018; Karras et al., 2020).

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 (Dias, 2018) 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 (Januzzi, 2008). 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" (Januzzi, 2008), 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. (Passos et al., 2014) 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.

To characterize ultrafine materials and average size consumption, Gontijo (Gontijo, 2018) performed previous image treatment using the Scanning Electronic Microscope (SEM). The particles in the images were digitized, scaled using software, classified by colors in size ranges (intervals), and, after classification, graphs of granulometric distributions were generated.

4 EDGE AI HARDWARE PRESENTATION

In this work, we decided to implement the solution using the SiPEED MAiX Dock board, displayed in Figure 2. Some performance numbers of the board are shown in Table 1. The work of Klippel et al. (Klippel et al., 2020) 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.

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

Figure 2: SiPEED M1 Dock.

Table 1: Embedded platform performance numbers.

Parameter	Characteristics
CPU	64-bit RISC-v processor and core
Chipset	K210 - RISC - V
Image Recognition	qvg at 60fps / vg at 30fps
Clock (GHz)	0.40
AI resources	KPU
OS/Language	uPython
Dimensions (mm)	60x43x5

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. Figure 3 displays the K210 block diagram (ken, 2018). The KPU is a general-purpose 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 1x1 and 3x3 convolution kernels;

- Support for any form of activation function;
- The maximum size of the supported neural network parameter for real-time work is from 5MiB to 5.9MiB.

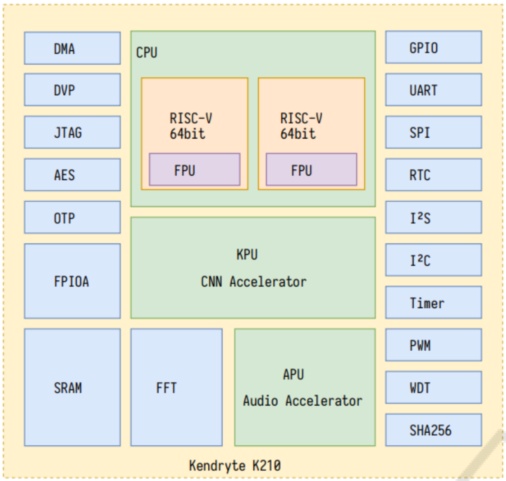

Figure 3: Block diagram of K210.

For training, the aXeleRaTe framework, a Keras-based framework for AI on the Edge, was used 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 4 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 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).

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 METHODOLOGY

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

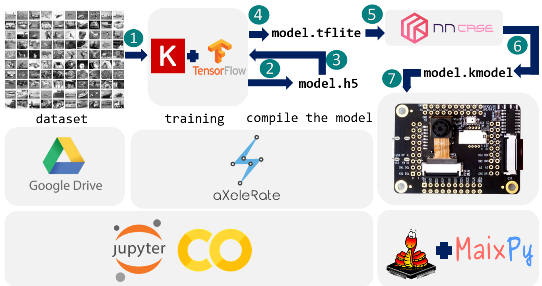

Figure 4: Training and compilation with aXeleRate.

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 generated a dataset with 1368 images to create a pilot appliance, containing 1140 for training and 228 for validation. The dataset has three different classes: quasi-particle, non-category, and empty. We also added 343 synthetic images produced on the bench-scale for the quasi-particle class training, as presented in Figure 5. 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.

Figure 5: Image of dataset developed on 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

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

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 training seasons, 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 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 *overall accuracy 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.

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

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

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

5.4 Edge AI Construction

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 224x224 resolution and storage on the SD card. The second to test the model from the storage data set previously stored on the SD card.

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 minutes, reaching an accuracy of 98.60%. Figure 6 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.

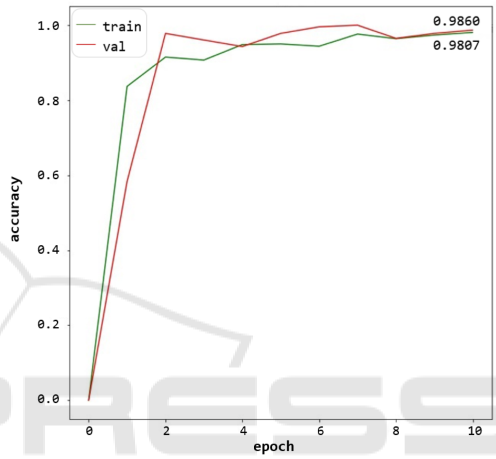

Figure 6: Training graph.

To validate the model, we created a dataset with 228 images. 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 quasi-particles). Table 2 displays the confusion matrix considering quasi-particles as the main class and Table 3 shows the performance indicators.

Table 2: Confusion matrix of model - Validation set.

		Predict		
		quasi_particle	non_category	empty
Real	quasi_particle	76	0	0
	non_category	1	74	1
	empty	0	0	76

Table 3: Trained Model Performance at Validation set.

Indicator	Value
precision	98,60%
recall	100%
F1	99,34%

The model accuracy was 98.70%. 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 Figure 7. For this matter, we tested the system using images from the three classes (quasi-particle, non-category, and empty). Table 4 displays the confusion matrix and Table 5 shows the performance indicators.

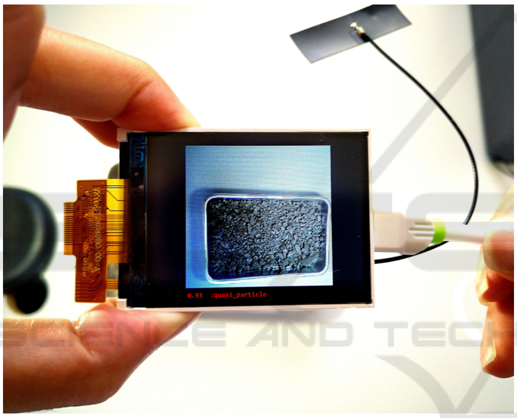

Figure 7: SiPEED MaiX Dock - Test Demonstration.

Table 4: Confusion matrix of model - Test set.

		Predict		
		quasi_particle	non_category	empty
Real	quasi_particle	7	2	1
	non_category	0	9	1
	empty	0	0	10

Table 5: Trained Model Performance at Test set.

Indicator	Value
precision	100,00%
recall	70,00%
F1	82,35%

In contrast to the value achieved in the validation set, the recall in the test set dropped to 70%, assessed from the SiPEED embedded system. This result indicates that the model had difficulties in testing positive for simulations of images similar to images in the industrial environment, as displayed in Figures 9 and 8, although for synthetic images with spaced particles was no difficulty, as shown in Figure 10.

The work of Klippel et al. (Klippel et al., 2020) 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.(Klippel et al., 2020) 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.

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

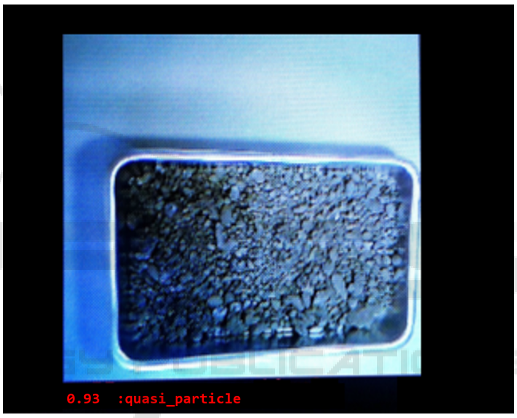

Figure 8: Example of recognition of quasi-particles simulating sampling in an industrial environment during the test using SiPEED.

Figure 9: Example of error in recognizing quasi-particles simulating sampling in an industrial environment during the test using SiPEED.

Figure 10: Example of recognition of quasi-particles with sample developed on a bench scale during the test using SiPEED.

7 CONCLUSION

In this work, we proposed an Edge AI appliance to classify images from the Hybrid Pellet Sinter (HPS) process from the steel industry. For this matter, we produced an application that aimed to detect trays containing quasi-particles, allowing them to differentiate from other objects and even empty trays. The results indicate the system feasibility and the possibility of loading the produced model into an Edge AI specialized hardware, although it needs improvement. The importance of obtaining the intervals of the particle size distribution of quasi-particles in the HPS process, its use of edge is shown as an advance, as currently, this process is carried out manually and in prolonged time intervals in the plants.

We proposed the appliance of a Convolutional Neural Network (CNN) embedded into an Edge Computing appliance. For this matter, the algorithm must analyze a dense scene, with problems like occlusion and complex background changes. Although there are broad applications of deep learning in dense scenes, there are still open issues to solve in the research process. The usage of specialized edge computing hardware enhances the performance of the solution on edge.

For testing this application, we used the SiPEED MaiX Dock board. This board has hardware and software infrastructure to enhance the development of Edge AI applications. The solution is cost- and resource-restrictive, given its specialized hardware to create edge deep learning applications. It also integrates with models created using the main DL frameworks.

Tests using SiPEED allow the detection of quasi-particles in synthetic images without difficulties, even with particles and their spaced distribution. However, tests with real images obtained some flaws, evidence by the drop in recall to 70%. There may have been overfitting during or training or, during the tests, had influences associated mainly with the brightness of the day, the occlusion between particles, color homogeneity, and overlap between objects. These problems have been reported in other works with previous advanced problems.

From the results obtained in this step, it was possible to raise new hypotheses for approaches to improve the deep learning algorithm to provide a granulometric range of the quasi-particles present in a sample. Thus, it is intended to evaluate other models for future work, such as supplying instances, making it possible to extract the histogram of the particle size distribution from the images containing quasi-particles. The development of an edge device with deep learning can bring significant benefits, both from process improvement and the insertion of steel processes in Industry 4.0.

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.

REFERENCES

(2018). K210 datasheet.

Chen, Z., Liu, L., Qi, X., and Geng, J. (2016). Digital mining technology-based teaching mode for mining engineering. *iJET*, 11(10):47–52.

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., and Zomaya, A. Y. (2020). Edge intelligence: the confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*.

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., and Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, pages 1–1.

Dias, Í. d. S. M. (2018). Sistema de controle granulométrico de pelotas de minério de ferro.

Gao, G., Gao, J., Liu, Q., Wang, Q., and Wang, Y. (2020). Cnn-based density estimation and crowd counting: A survey. *arXiv preprint arXiv:2003.12783*.

Gontijo, M. D. (2018). Análise granulométrica por imagem de amostras ultrafinas. *Revista Engenharia de Interesse Social*, 1(3).

Januzzi, A. (2008). Análise da aglomeração a frio no processo hps (hybrid pelletized sinter) com ênfase nas matérias-primas envolvidas.

Karras, K., Pallis, E., Mastorakis, G., Nikoloudakis, Y., Batalla, J. M., Mavromoustakis, C. X., and Markakis, E. (2020). A hardware acceleration platform for ai-based inference at the edge. *Circuits, Systems, and Signal Processing*, 39(2):1059–1070.

Keresztes, B., Abdelghafour, F., Randriamanga, D., da Costa, J.-P., and Germain, C. (2018). Real-time fruit detection using deep neural networks. In *14th International Conference on Precision Agriculture*.

Kinnunen, P. H.-M. and Kaksonen, A. H. (2019). Towards circular economy in mining: Opportunities and bottlenecks for tailings valorization. *Journal of Cleaner Production*, 228:153–160.

Klippel, E., Oliveira, R., Maslov, D., Bianchi, A., Silva, S. E., and Garrocho, C. (2020). 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*, pages 97–102, Porto Alegre, RS, Brasil. SBC.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Li, E., Zeng, L., Zhou, Z., and Chen, X. (2019). Edge ai: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications*, 19(1):447–457.

Lin, X., Li, J., Wu, J., Liang, H., and Yang, W. (2019). Making knowledge tradable in edge-ai enabled iot: A consortium blockchain-based efficient and incentive approach. *IEEE Transactions on Industrial Informatics*, 15(12):6367–6378.

Mardonova, M. and Choi, Y. (2018). Review of wearable device technology and its applications to the mining industry. *Energies*, 11(3):547.

Mazzia, V., Khaliq, A., Salvetti, F., and Chiaberge, M. (2020). Real-time apple detection system using embedded systems with hardware accelerators: An edge ai application. *IEEE Access*, 8:9102–9114.

Ohbuchi, E. (2018). Low power ai hardware platform for deep learning in edge computing. In *2018 IEEE CPMT Symposium Japan (ICSJ)*, pages 89–90. IEEE.

Passos, L. A. S., Moreira, J. L., Jorge, A., and Cavalcante, M. V. S. (2014). 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.

Robben, C. and Wotruba, H. (2019). Sensor-based ore sorting technology in mining—past, present and future. *Minerals*, 9(9):523.

Saleem, M. H., Potgieter, J., and Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants*, 8(11):468.

Shibuta, Y., Ohno, M., and Takaki, T. (2018). Computational metallurgy: Advent of cross-scale modeling: High-performance computing of solidification and grain growth (adv. theory simul. 9/2018). *Advanced Theory and Simulations*, 1(9):1870020.

Sinoviev, V. V., Okolnishnikov, V. V., Starodubov, A. N., and Dorofeev, M. U. (2016). Approach to effectiveness evaluation of robotics technology in mining using discrete event simulation. *International Journal of Mathematics and Computers in Simulation*, 10:123–128.

Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., and Chen, M. (2019). In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 33(5):156–165.

Zhang, Q., Liu, Y., Gong, C., Chen, Y., and Yu, H. (2020). Applications of deep learning for dense scenes analysis in agriculture: A review. *Sensors*, 20(5):1520.

Zhou, X., Fang, B., Qian, J., Xie, G., Deng, B., and Qian, J. (2019a). Data driven faster r-cnn for transmission line object detection. In *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*, pages 379–389. Springer.

Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., and Zhang, J. (2019b). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762.

Zobnin, N. N., Torgovets, A. K., Pikalova, I. A., Yussupova, Y. S., and Atakishiyev, S. A. (2018). Influence of thermal stability of quartz and the particle size distribution of burden materials on the process of electrothermal smelting of metallurgical silicon. *Oriental Journal of Chemistry*, 34(2):1120–1125.