APPLICATION OF DEEP LEARNING FOR ANALYSIS OF CRACKS IN PELLET FALLING TESTS

Marconi J.H. Magnani[1], Thyago R. Souza[1], Jorge J. F. Filho[1], Marco A. D. S. L. Cuadros[1]

[1]*Graduate Program in Automation and Control, Federal Institute of Espírito Santo*
ES-010, Km-6,5, Manguinhos, 29173-087, Serra/Espírito Santo, Brazil
marconi@in9automacao.com.br,thyago@in9automacao.com.br,
jorge@in9automacao.com.br,marcoantonio@ifes.edu.br

Abstract. Iron ore pellets are a prime input for iron production. Therefore there is a need for a rigorous control of the quality of the pellets to apply them in the industrial process. The pellets are degraded due to impacts caused by their handling or transport systems. As a result of these degradations many pellet shipments reach the customer with a proportion of cracks. Laboratory drop test trials are required on wet raw pellets to assess their resistance to the various drops they suffer in the industrial process. Currently the drop test is performed manually, where the whole test process, from pellet manipulation and data collection, depends on human action. The present work aims at the application of Deep Learning to carry out the analysis of pellet cracks, pellet segmentation is initially presented in this article. A network of light deep learning was designed, generating a data set of the pellet drop test for training the network for pellet classification. This network will be applied in the autonomous prototype for the drop test, a technological innovation that is being developed by the Research and Automation Group (GAIn), of the Federal Institute of Espírito Santo, located in the Municipality of Serra, for the analysis of cracks in the pellets.

Keywords: Pellets, Industrial Process, Drop Test, Deep Learning, Pellet Cracks.

1 Introduction

Currently cars, trucks, bicycles, airplanes, appliances and most of the products we use in our daily lives are made from the steel produced in the steel mills. To obtain it, an important raw material used is the pellet. Pelletizing is a process responsible for pellet production.

Iron ore pellet is one of the main inputs in the primary iron production stage within the production route of steel, as it presents physical, chemical and metallurgical companies more favorable to reduction operations compared to other raw materials such as iron ore granulate and the sinter. Among these characteristics, cite the narrowest size distribution, the high porosity, low ignition loss, the highest content of iron, among others, as Meyer [1] mentions.

Strict control over the quality of pellets, a control method used is the drop test, performed on raw wet pellets, this test allows to evaluate the resistance of the raw wet pellets to several falls that they suffer from the disc of pelletizing to the mobile grid oven.

Each pellet is individually released from a height of approximately 45cm several times until the same present some crack. Thus, the number of falls that pellet endured will be the resilience value (ability to return to the natural state after an unusual situation). After several tests the average value is reported as a result.

Currently the test of the drop test is carried out in a manual, where the entire testing process, from handling pellets and obtaining the data, depends on an action human.

Neural networks and deep learning techniques are being widely used in the most diverse areas and segments as, for example in medicine where they are used for determination of kidney diseases by analyzing the most diverse characteristics of this organ, said Raju, Rao and Rao [2].

In recent years, Deep Learning has revolutionized the field of machine learning, for computer vision in

Special. In this approach, an artificial neural network (RNA) deep (multilayer) is trained, usually in a manner supervised using backpropagation. Are required large amounts of labeled training examples, but the accuracy of the resulting classification is really impressive, sometimes surpassing humans as Tavanaei, Ghodrati, Kheradpisheh, Masquelier and Maida [3]. Numerous works related to detection using Deep Learning can be found in the literature in Duan, Liu, Wu and Mao [4] a lightweight network of deep network learning was designed U-net to automatically detect image pellets and obtain pellet contour probability maps, deep learning techniques were used in medicine for determining kidney disease by analyzing the more diverse characteristics of this organ, performed by Raju, Rao and Rao [2].

This article proposes the use of Intelligence techniques Artificial for a technological innovation project, where it will be A deep convolutional network for analysis and segmentation of pellets.

This article is divided into 5 chapters. This, chapter 1, which contextualizes the problem to be solved, exposes the justification and importance of work. Chapter 2 on the process pelletizing process and the quality test called (drop test), a test performed manually with the objective of check the number of falls that the ball can take. The chapter 3 that presents the theoretical framework that supports this research. Chapter 4, which explains the development of work and ends with chapter 5 with the conclusion of the work.

2 Pelletizing

Bridges, cars, planes, bicycles, appliances and large part of the products we use in our daily lives are made from steel produced in steel mills. To obtain it, an important raw material used is pellet, whose processing from ultrafine fractions of ore iron, takes place in our pelletizing plants, according to Vale [5].

The production system for these small spheres begins with the extraction of iron ore in Minas Gerais. The fine iron ore, called pellet-feed, arrives at the yards of the producing units from the mines. In the courtyards are piles are formed which are subsequently recovered and conveyed on belts for the grinding process. In parallel, the yards receive inputs, such as limestone, that will be added to the ore. In grinding, the ore is ground with water, forming a pulp classified by hydrocyclones (equipment for separating solid and liquid) and sent to the thickener, where it is sedimented and, in then forwarded to homogenizing tanks, as reported by Vale [5], as shown in Figure 1.

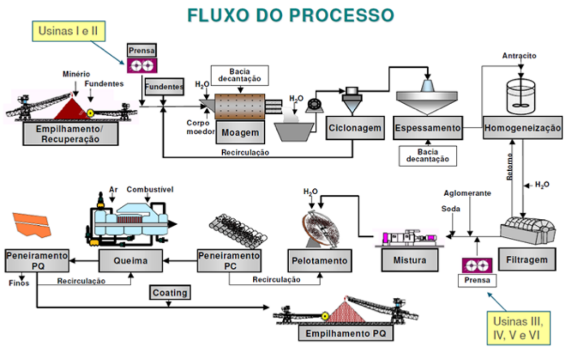


Figure 1. Pelletizing Process Flow. Source: Vale (2014)

The quality of the pellets is verified through several tests, and one of them is the drop test, performed on pellets raw wet, this test allows to evaluate the resistance of the damp raw pellets to the various falls that they suffer from the pellet disc to the grid. Each ball is released individually from a height of approximately 45cm several times.

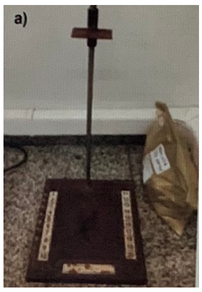


Figure 2. Drop Test

Figure 2 shows the equipment used for the test.

3 Smart System

3.1 Neural Networks

Artificial Neural Networks are simultaneous systems composed of simple processing units, neurons, that calculate certain mathematical functions, linear or non-linear, according to Braga, Ludemir and Carvalho [6].

According to Sugomori [7] while the other methods use probability and statistics, the neural network algorithms seek to imitate the structures of the human brain.

3.2 Convolutional Neural Network

In the context of deep neural networks, convolutional networks have been shown to be very efficient in image processing applications. According to Ponti [8], deep learning methods are now the state of the art in many machine learning problems, in particular in classification problems.

Also according to Ponti [8], CNNs (Convolutional Neural Networks), are the models of Deep Learning networks most known and used today. What characterizes this type of network is that convolutional layers are placed, which process the inputs considering local fields, among other devices, such as pooling, which reduces the spatial dimension of the inputs. Figure 3 shows the application of convolution in regions of an input image.

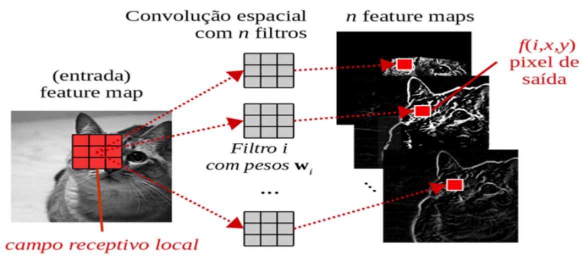


Figure 3. Space Convolution. Source: Ponti (2017)

In the convolutional layer each neuron is a filter applied to an input image and each filter is a matrix of weights.

In Figure 4, it is possible to observe the application of two layers of convolution in an RGB image, that is, an image with three dimensions. In the first layer, 4 5x5x3 filters are applied, producing 4 feature maps, and then another convolutional layer with 5 3x3x4 filters, generate new feature maps, as detailed by Ponti [8].

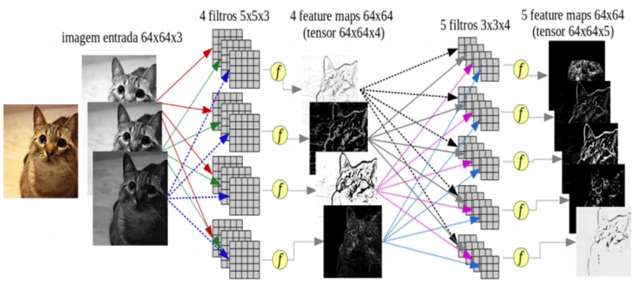


Figure 4. CNN with two convolutional layers. Source: Ponti (2017)

4 Development

For the development of the system, a pellet drop test data set was created for training the deep learning network, that is, images of the pellets were obtained in the course of the tests to create the data set, initially approximately 50 test figures were used fall, some had fractures and others did not, techniques were used to treat the images as shown in Figure 5 for network training.

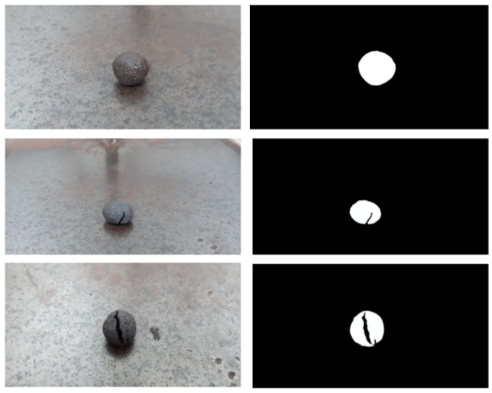


Figure 5. Image processing

A lightweight deep learning network was designed as shown in Figure 6.

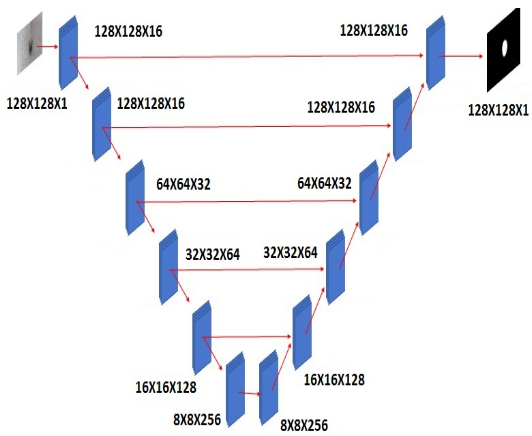
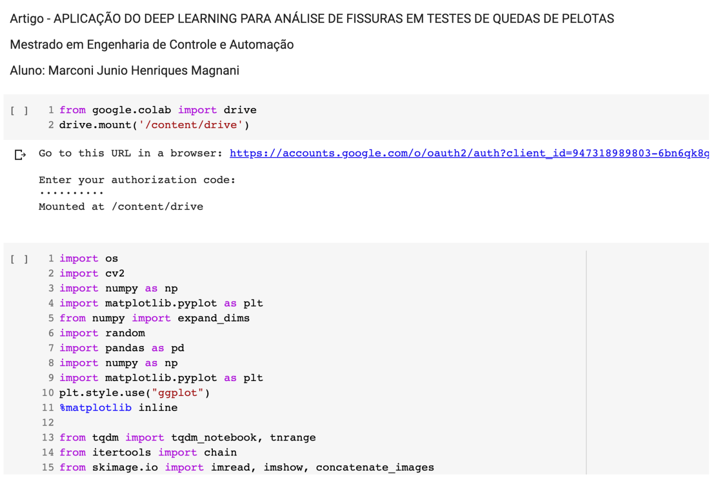


Figure 6. Network structure used

To classify the pellets, a light deep learning network from the U-net network was used to automatically detect pellets from images and obtain the pellet contour probability maps as shown in Figure 7, Figure 8 and Figure 9.



Artigo - APLICAÇÃO DO DEEP LEARNING PARA ANÁLISE DE FISSURAS EM TESTES DE QUEDAS DE PELOTAS

Mestrado em Engenharia de Controle e Automação

Aluno: Marconi Junio Henriques Magnani

```
[ ]
from google.colab import drive
drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8q

Enter your authorization code:
··········
Mounted at /content/drive

[ ]
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from numpy import expand_dims
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("ggplot")
%matplotlib inline

from tqdm import tqdm_notebook, tnrange
from itertools import chain
from skimage.io import imread, imshow, concatenate_images
```

Figure 7. Program code

Figure 7 represents the program code, the tool used to develop the program was Google Collaboratory, known commercially as Google Colab.

```python
def get_unet(input_img, n_filters=16, dropout=0.5, batchnorm=True):
    # contracting path
    c1 = conv2d_block(input_img, n_filters=n_filters*1, kernel_size=3, batchnorm=batchnorm)
    p1 = MaxPooling2D((2, 2)) (c1)
    p1 = Dropout(dropout*0.5)(p1)

    c2 = conv2d_block(p1, n_filters=n_filters*2, kernel_size=3, batchnorm=batchnorm)
    p2 = MaxPooling2D((2, 2)) (c2)
    p2 = Dropout(dropout)(p2)

    c3 = conv2d_block(p2, n_filters=n_filters*4, kernel_size=3, batchnorm=batchnorm)
    p3 = MaxPooling2D((2, 2)) (c3)
    p3 = Dropout(dropout)(p3)

    c4 = conv2d_block(p3, n_filters=n_filters*8, kernel_size=3, batchnorm=batchnorm)
    p4 = MaxPooling2D(pool_size=(2, 2)) (c4)
    p4 = Dropout(dropout)(p4)

    c5 = conv2d_block(p4, n_filters=n_filters*16, kernel_size=3, batchnorm=batchnorm)

    # expansive path
    u6 = Conv2DTranspose(n_filters*8, (3, 3), strides=(2, 2), padding='same') (c5)
    u6 = concatenate([u6, c4])
    u6 = Dropout(dropout)(u6)
    c6 = conv2d_block(u6, n_filters=n_filters*8, kernel_size=3, batchnorm=batchnorm)

    u7 = Conv2DTranspose(n_filters*4, (3, 3), strides=(2, 2), padding='same') (c6)
    u7 = concatenate([u7, c3])
    u7 = Dropout(dropout)(u7)
    c7 = conv2d_block(u7, n_filters=n_filters*4, kernel_size=3, batchnorm=batchnorm)

    u8 = Conv2DTranspose(n_filters*2, (3, 3), strides=(2, 2), padding='same') (c7)
```

Figure 8. Designed U-Net deep learning network

Figure 8 represents the configuration of the U-Net network, where all network parameters were defined.

Model: "model_2"

Layer (type)	Output Shape	Param #	Connected to
img (InputLayer)	(None, 128, 128, 1)	0	
conv2d_20 (Conv2D)	(None, 128, 128, 16)	160	img[0][0]
batch_normalization_19 (BatchNo	(None, 128, 128, 16)	64	conv2d_20[0][0]
activation_19 (Activation)	(None, 128, 128, 16)	0	batch_normalization_19[0][0]
conv2d_21 (Conv2D)	(None, 128, 128, 16)	2320	activation_19[0][0]
batch_normalization_20 (BatchNo	(None, 128, 128, 16)	64	conv2d_21[0][0]
activation_20 (Activation)	(None, 128, 128, 16)	0	batch_normalization_20[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 64, 64, 16)	0	activation_20[0][0]
dropout_9 (Dropout)	(None, 64, 64, 16)	0	max_pooling2d_5[0][0]
conv2d_22 (Conv2D)	(None, 64, 64, 32)	4640	dropout_9[0][0]
batch_normalization_21 (BatchNo	(None, 64, 64, 32)	128	conv2d_22[0][0]
activation_21 (Activation)	(None, 64, 64, 32)	0	batch_normalization_21[0][0]
conv2d_23 (Conv2D)	(None, 64, 64, 32)	9248	activation_21[0][0]
batch_normalization_22 (BatchNo	(None, 64, 64, 32)	128	conv2d_23[0][0]
activation_22 (Activation)	(None, 64, 64, 32)	0	batch_normalization_22[0][0]
max_pooling2d_6 (MaxPooling2D)	(None, 32, 32, 32)	0	activation_22[0][0]

Figure 9. Parameters of the designed U-Net deep learning network.

A network training was carried out using the data set created, in this step the number of times was defined, and network parameters in order to obtain the greatest possible amount of correctness, always observing the accuracy of the test results. After training, the result of accuracy as a goal will be over 87% as shown in Figure 10.

```
Epoch 00020: val_loss did not improve from 0.29790
Epoch 21/200
432/432 [==============================] - 2s 5ms/step - loss: 0.2883 - acc: 0.9537 - val_loss: 0.3294 - val_acc: 0.9495

Epoch 00021: val_loss did not improve from 0.29790
Epoch 22/200
432/432 [==============================] - 2s 5ms/step - loss: 0.2858 - acc: 0.9540 - val_loss: 0.3249 - val_acc: 0.9500

Epoch 00022: ReduceLROnPlateau reducing learning rate to 1e-05.

Epoch 00022: val_loss did not improve from 0.29790
Epoch 23/200
432/432 [==============================] - 2s 5ms/step - loss: 0.2876 - acc: 0.9537 - val_loss: 0.3210 - val_acc: 0.9504

Epoch 00023: val_loss did not improve from 0.29790
Epoch 24/200
432/432 [==============================] - 2s 4ms/step - loss: 0.2854 - acc: 0.9541 - val_loss: 0.3170 - val_acc: 0.9508

Epoch 00024: val_loss did not improve from 0.29790
Epoch 25/200
432/432 [==============================] - 2s 5ms/step - loss: 0.2871 - acc: 0.9539 - val_loss: 0.3137 - val_acc: 0.9512

Epoch 00025: val_loss did not improve from 0.29790
Epoch 26/200
432/432 [==============================] - 2s 5ms/step - loss: 0.2859 - acc: 0.9540 - val_loss: 0.3106 - val_acc: 0.9515

Epoch 00026: val_loss did not improve from 0.29790
Epoch 00026: early stopping
```

Figure 10. Network training.

Figure 11 represents the network learning curve after training the network, to perform the training of the network it was considered the use of some training optimization tools such as early stop and the reduction of the network learning rate. The early stop method is a method that retains the best training result within a sequence of subsequent results, it is used to prevent overfit training, in which case a 12-season early stop was used which means that after the best result will occur 12 seasons of worst results and will end the training. The reduction in the learning rate of the network was used so that in the event of a worsening trend in training, the learning rate is adjusted for an attempt to converge the training. In this case, a rate adjustment was used every two sequential times of worse results than the best learning outcome. When the training started, the loss (loss) and the loss of validation (loss) started high as expected, the loss curve was characterized as asymptotic and stabilizing, the validation loss had a variable behavior, being that in times 7 , 19 and 22 the rate was adjusted according to the worsening of the response to training, the best result was obtained in the 2000 season, it was not necessary to stop early, as shown in Figure 11.

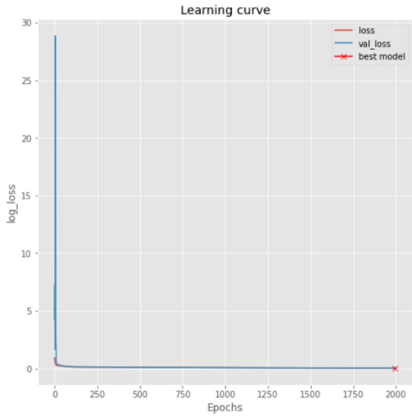


Figure 11. Network learning curve.

After training the network, the images were predicted in order to verify the quality of the training, Figure 12 represents the result of two predictions of incoming images in the trained U-Net network. The first image is the input image of the network, the second image is the image of the input image mask, that is, it is the image with which we will compare with the image of the network output. The third image is the prediction output from the network and the fourth image is the prediction output with binary pixel values (values 0 or 1).

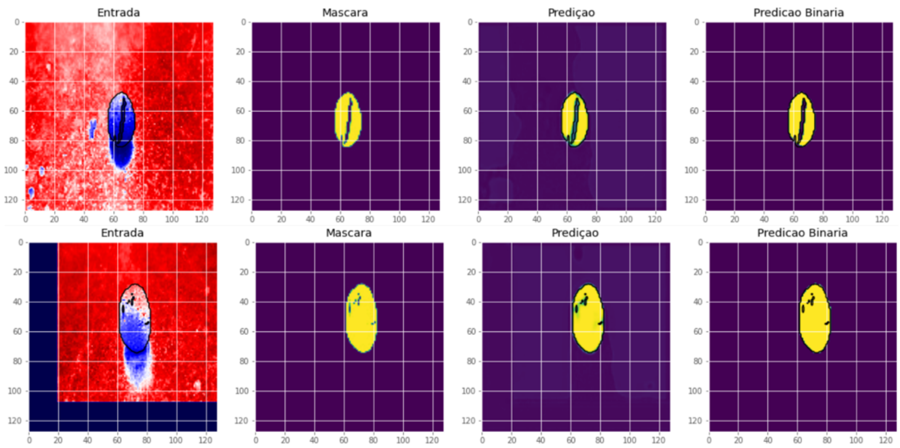


Figure 12. Test prediction.

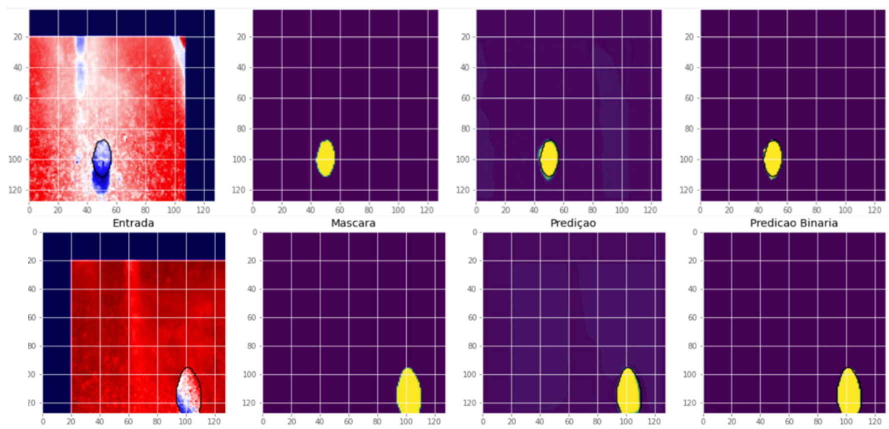


Figure 13. Validation test.

Figure 13 represents the validation of the network training, where 48 images were used that are not part of the network training data for network validation, and the results can be observed in three images as shown in

Figure 13, entry, mask, prediction and binary prediction.

```
import sklearn.metrics as metrics
from skimage import measure

med=0
for a in range(len(preds_val)):
  s = measure.compare_ssim((preds_val[a][:,:,0]> 0.5).astype(np.uint8), (y_valid[a][:,:,0]> 0.5).astype(np.uint8))
  med=med+s

med = med/len(preds_val)
print(med)
```

```
0.9999555360027715
```

Figure 14. Checking the model.

To compare the masks generated by the mapped masks, the method of comparison by structural similarity was used, the same method was applied and compared the image from the prediction with the mask created, as a result the result was 99.9955%, that means that both images are approximately 100% identical, validating the quality of the network training.

5 Conclusions

The projected network met the expectations, we used an accuracy target above 87% and the trained network showed an average accuracy of 97% during training and greater than 99% for validation. When analyzing the result for a prediction of an image not used in the training, it was found that the quality of the prediction exceeds the expectations and goals created at the beginning of the project.

For future work, a dense network will be applied at the exit of the U-net to classify good or bad pellets where real-time images will be collected that will be used as input to the network, for validation and application in the autonomous prototype.

This network will be applied in the autonomous prototype for the drop test, a technological innovation that is being developed by the Research and Automation Group (GAIn), of the Federal Institute of Espírito Santo, located in the Municipality of Serra, to analyze the cracks in the pellets.

Acknowledgements. We thank God first, because without Him, nothing would be possible, we also thank our families and friends.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Meyer, K., 1980, Pelletizing of Iron Ores, Springer-Verlag Berlin, Heidelberg, Germany.
[2] Raju, P., Rao, V. M., & Rao, B. P. (2018). Grey Wolf Optimization-Based Artificial Neural Network for Classification of Kidney Images. Journal of Circuits, Systems and Computers, 27(14), 1–21. https://doi.org/10.1142/S0218126618502316.
[3] Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. Neural Networks, 111, 47–63. https://doi.org/10.1016/j.neunet.2018.12.002.
[4] Duan, J., Liu, X., Wu, X., & Mao, C. (2019). Detection and segmentation of iron ore green pellets in images using lightweight U-net deep learning network. Neural Computing and Applications, 8. https://doi.org/10.1007/s00521-019-04045-8.
[5] VALE. (2014). Entenda como funciona o processo de pelotização em nossas usinas. Retrieved from http://www.vale.com/brasil/PT/aboutvale/news/Paginas/entenda-funciona-processo-pelotizacao-usinas.aspx
[6] BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. Redes Neurais Artificiais: teoria e aplicações. 1 ed. Rio de Janeiro: LTC, 2000.
[7] SUGOMORI, Y. et al. Deep Learning: Practical Neural Networks with Java. [S.l.]: Packt Publishing Ltd, 2017.
[8] Ponti MA, Costa GBP da. Como funciona o deep learning [Internet]. In: Tópicos em gerenciamento de dados e informações 2017. Uberlândia: SBC; 2017. Available from: http://sbbd.org.br/2017/wp-content/uploads/sites/3/2017/10/topicos-em-gerenciamento-de-dados-e-informacoes-2017.pdf