2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)

Automatic Recognition of Electrical Grid Elements using Convolutional Neural Networks

L. A. Z. Silva*, V. F. Vidal†, M. F. Silva‡,
M. F. Santos§, A. L. Carvalho¶, A. S. Cerqueira||, L. M. Honório**,
H. B. Rezende††, J. M. S. Ribeiro‡‡, A. A. N. Pancoti[x], B. A. Regina[xi]

Centro Federal de Educação Tecnológica de Minas Gerais,
CEFET-MG, Leopoldina, MG, 36700-000, Brazil
murillo.ferreira@cefetmg.br§

Juiz de Fora Federal University,
UFJF, Juiz de Fora, MG, 36036-900, Brazil
luiz.zillmann@engenharia.ufjf.br*, vinicius.vidal@engenharia.ufjf.br†, mathausfsilva@hotmail.com‡,
alexandre.carvalho@mrs.com.br¶, augusto.santiago@ufjf.edu.br||, leonardo.honorio@ufjf.edu.br**,
henrique.rezende@engenharia.ufjf.br††, joao.ribeiro@engenharia.ufjf.br‡‡, antonio.pancoti@engenharia.ufjf.br[x],
almeida.bruno@engenharia.ufjf.br[xi]

***Abstract*—Due to the extensive proportions of Brazilian railways, there is a high demand for remote and automatic diagnose tools. This work proposes a scene selection method using Deep Learning techniques, namely Convolutional Neural Networks (CNN), to recognize the poles, which gathers objects of interest to be inspected in the railway power network. Videos were obtained through the railway and the data divided and preprocessed for the network training and testing. A VGG network architecture served as a starting point, and after exhaustive search and comparisons of many techniques, two network topologies are presented and compared in field tests. The results yield more than 93% efficiency for both proposed topologies.**

***Keywords*—Machine Learning, Deep Learning, Convolutional Neural Networks, Railway Maintenance.**

I. Introduction

Railways usually need frequent preventive maintenance [1], with acoustic or thermographic equipment [2][3]. Automatic inspection aims to help with this issue. The data acquisition is made by cameras and the videos and images captured are analyzed in order to identify desired features. Elements like crosshead, isolator and fuse switch are examples of equipment that, if damaged, can cause short-circuit and current leakage, which leads to more damages and danger to the line and people around [4].

Image processing usually demands reasonably high computational cost and is susceptible to errors. In the works of [5] and [6], a characteristic is inserted in the environment to help identify the location and gather information during the aircraft flight with low processing time and resources. The technique makes use of geometric shapes recognition at most.

Complex computational problems have been a challenge for hardware resources even nowadays, and techniques such as the polymorphic agents [7] arise in the literature, combining optimization and programming methods, to coordinate task scheduling and solving. According to LeCun et al.[8], conventional machine learning techniques were limited when it was necessary to process raw data, demanding considerable work and expertise from the project planner during the feature extraction step. To overcome this barrier, Chen et al. [9] makes use of convolution neural networks (CNN) combined with naive Bayes descriptor to detect cracks in nuclear power plants. Singh et al.[10] and Silva et al.[11] also applied CNN. The first, fusing sensor data and mapping footprints, while the second one applied it in cattle marks recognition.

In this paper, it was proposed a solution to identify poles using convolutional neural networks. This deep learning method has shown to be robust when compared to other state-of-art techniques in classifying images, since, the deeper the information follows in the network layers, more detailed and abstract the features are [12].

Within the literature, it is possible to find related papers which propose the recognition of elements automatically. In [13], the work goal is finding damage and flaws in rails using CNN, emphasizing the robust network technique over an extensive database. In [3], the subject is the quality monitoring of catenary supports in Chinese railway network, which may have billions of elements to be monitored using CNN. Both papers focus on the discrimination of elements in several classes for each characteristic of interest.

The paper is organized as follows: Section II presents the proposed method to detect and locate the element of interest. Section III presents the initial setup to solve the problem, results obtained with training the proposed topologies and element recognition in videos not used in training. Finally, Section IV brings the conclusions on the proposed approach.

II. Proposed Methodology

This section presents the proposed methodology, illustrated by the Fig. 1, in order to detect and localize elements of interest. The first block represents the image acquisition step; the second one indicates the database formation to accomplish the training and validation of the proposed method, followed by the image treatment, generating input for the convolutional neural network. The CNN output indicates whether the entrance image is a pole or not.

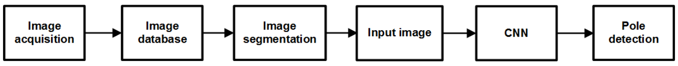

Fig. 1. Bloc diagram of the proposed methodology.

A. Acquisition and image database formation

This work proposes the acquisition of videos for processing by an RGB camera fixed in front of a road-rail vehicle. The database contains videos from which the images are taken and used in all steps of this approach: training, validation, and tests.

B. CNN project and image treatment

In order to achieve a better system performance, two network topologies are proposed, evaluating their results against a VGG-like network [14], as is proposed in [12]. These topologies are trained and validated using frames from the database described before and tested with videos dedicated to the testing phase.

For the network training and validation, it is defined a set of images containing poles, called positive images. Moreover, the second set of images is created containing all elements but poles, called negative. For the test step, several different videos were utilized during the CNN project.

Besides that, a segmentation of the positive images was necessary, since poles are positioned vertically in the images, with a rectangular occupation. The database contains RGB images, meaning that they have three matrices of pixels, one for each color channel. Since the network input must have fixed dimensions, there is a need of preprocessing the input, preparing images to be used. Input size is, then, set to 250x35 for each color channel, which means, 250x35x3.

Regarding the proposed CNN topologies, one of them contains more convolutional layers (denser network), focusing on false-positives reduction. The other one focuses on processing speed.

The less dense one, denominated as the fast network, as illustrated by the Fig. 2, has four convolutional layers, followed by a pooling layer after each one [15]. Besides that, the dropout technique, represented by the blue arrow in the figure, was applied in order to avoid over-fitting, as mentioned in [12] and [8].

Fig. 3 illustrates the dense network, also called deep network, since it has more convolutional and, consequently, more pooling layers. As in the previous one, the dropout technique has been applied before the output, also represented as a blue arrow.

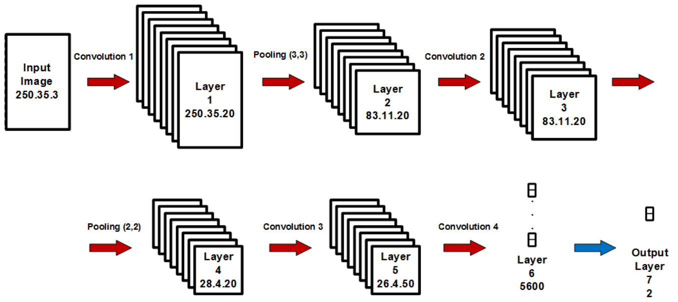

Fig. 2. Fast network topology.

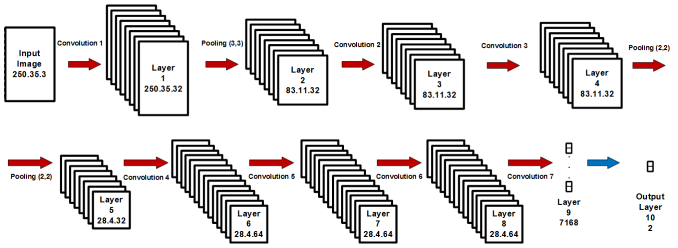

Fig. 3. Deep network topology.

C. Pole detection

As seen in Figs. 2 and 3, the output layer contains two elements. One of them presents the probability for the input to be a pole, while the other one the probability for anything else but a pole.

Therefore, the validation set is presented to the network, evaluating whether the training was correct or not. Once the training and validation are finished and the topology defined, different video sets are presented to the network, aiming a better performance from the system. In this step, different scales of moving windows are passed through every frame of the video, and their gathered content processed as mentioned in Section II-B, to then be delivered to the network and classified as a pole or not.

III. Experimental Results

In this section, the results obtained by the proposed methodology in tests realized in the state of Minas Gerais, Brazil, are presented.

A. Camera and Processing

Image acquisition was made using MAKO Gigabit Ethernet camera (Fig. 4), in RGB format. After the acquisition, videos are processed as a set of images, separated into groups for training and tests. Every frame represents a scene from where the data are extracted and delivered to the algorithm.

The results of this paper were obtained using a computer with i7-7^{th} generation processor, Ubuntu 16.04 operational system, CUDA-9.0 parallel processing and NVidia 1060 graphics card.

B. Feature Extraction

According to [12], the images delivered to the CNN should be square, which means have equal dimensions, in order to use

Fig. 4. Camera MAKO Gigabit Ethernet and Acquisition support.

matrix algebra properties that lead to better computational performance. Common sizes are 32x32, 64x64, 96x96, 224x224, 227x227 and 229x229. However, that would be a limiting factor to the application since the pole occupies a rectangular region in the image. Testing consecrated networks, like Lenet [16], with rectangular images, gave better results than with squared ones. Therefore, a rectangular segmentation was taken to proceed.

Given a set of more than 1200 frames, 83 pole images were extracted with good quality, eliminating most background features as possible, segmenting them to have the input size of 250x35x3. For negative elements, more than 600 images were extracted due to the variety of possible negative elements. The same segmentation was made. Fig. 5 shows pole images, where the right one was a better choice to solve the problem presented avoiding background and sky as much as possible.

Fig. 5. At left, background contaminated pole image. In the middle, pole image impaired by the sky. At right, image focused on the pole.

C. Proposed topologies training process

For supervised training of the topologies proposed in II, the dataset was divided following the proportion of 75% for training and 25% for validation.

The chosen approach was to minimize a loss function applying stochastic gradient descent, using momentum to increase gradient influence if it stood in the same direction in the next iteration, and decrease otherwise.

Nesterov acceleration was also used for training [17], which can be considered as a correction of the momentum.

To achieve the proposed network architectures several modifications were made and reported, such as the number of convolutional layers, changes in activation volumes in convolutional layers, different sizes of kernels, insertion of dropout and batch normalization techniques. Some techniques come in handy when computing time is concerned, as it is shown in [18], to improve efficiency in online processes.

Incrementing the deep network with batch normalization after the first two convolutional layers, illustrated by Fig. 6, has shown an unexpected behavior, according to the literature [12], in the validation loss values, even achieving a good result posteriorly. This behavior justifies the non-application of this technique in the proposed networks.

Fig. 7 shows the validation loss function smoothly decreasing, expected behavior obtained with the application of momentum and Nesterov acceleration, reaching values lower than 0,2 for the validation set. The accuracy increases asymptotically to 100%.

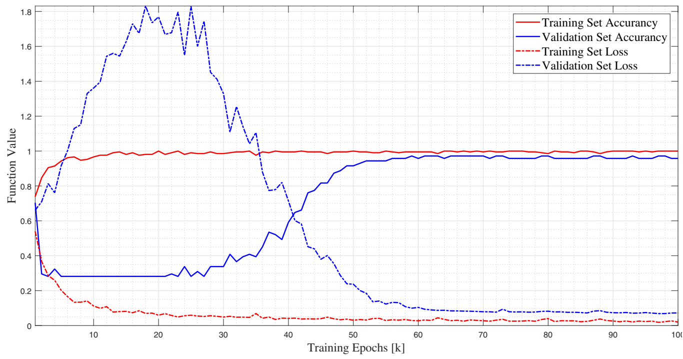

Fig. 6. Training evolution for deep network with batch normalization, showing unexpected behavior in validation loss.

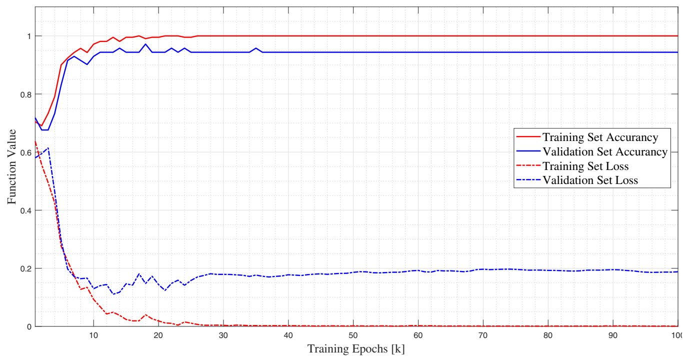

Fig. 7. Training evolution for the deep network without batch normalization, with validation loss decreasing smoothly.

The Table I shows mean values over accuracy percentage, for the validation set over poles (positives) and non-poles (negatives) classes, for the evaluated networks. It is shown the superiority obtained by the proposed topologies, more noticeable with positive images. A comparison between fast and deep networks shows, as expected, that the second one achieved more accuracy during validation, with a higher computational cost.

TABLE I
ACCURACY MEAN FOR VALIDATION SET.

Network	Positives	Negatives	Mean
mini-VGG	0,47	0,80	0,64
Fast	0,93	0,99	0,96
Deep	0,97	0,99	0,98

D. Performance

In order to evaluate topology performance, it was used a rectangular sliding window through every frame within the test video (Fig. 8), using the image selected by the window to be delivered to the CNN, varying the size of the window at every round sweep end. It is important to emphasize that the image obtained by the sliding window is preprocessed to conform with the CNN input size. Window score (in green) and processing time (in yellow) for that frame were accounted. Fig. 9 shows the result for a random frame.

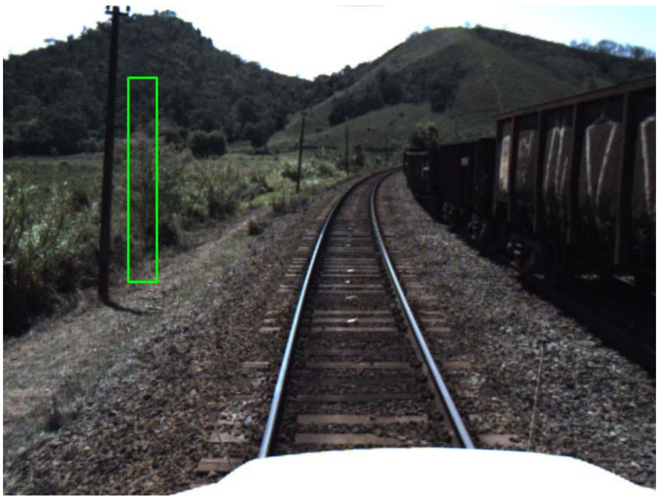

Fig. 8. Sliding window in a random frame.

In each collected video for the test step, for each architecture, were evaluated mean processing time, lost poles, found poles (called positives) and false positives (when a non-pole is marked as a pole). Mean and standard deviation of the success score for each video were calculated. The results for video 1, containing 27 poles and 994 frames, are presented in Tables II and III. For video 2, with 10 poles and 1105 frames, the results are presented in Tables IV and V. It is important to emphasize that the same pole appears in several frames of the video, being accounted many times.

As shown in Tables III and V, the time presented by the proposed fast topology is more than 50% lower when compared with the mini-VGG network, even with more convolutional layers. This can be explained by the absence of batch normalization between the CNNs' layers, which not only reduced the time but increased the accuracy rate with this database as well. A comparison between fast and deep network shows that the processing time of the first one is about two-thirds of the

Fig. 9. Finding a pole during a sweep in a test video, with time shown in yellow in hundredths of seconds.

TABLE II
RESULTS OF LOST POLES, POSITIVE FRAMES AND FALSE POSITIVES FOR VIDEO 1.

Network	Lost Poles	Positives	False Positives
mini-VGG	12	70	805
Fast	4	166	73
Deep	4	204	194

TABLE III
RESULTS OF PROCESSING TIME, OUTPUT MEAN AND STANDARD DEVIATION FOR VIDEO 1.

Network	Time (s)	Output mean	Standard Deviation
mini-VGG	0,25	0,951	0,021
Fast	0,10	0,955	0,026
Deep	0,15	0,977	0,027

TABLE IV
RESULTS OF LOST POLES, POSITIVE FRAMES AND FALSE POSITIVES FOR VIDEO 2.

Network	Lost Poles	Positives	False Positives
mini-VGG	1	161	783
Fast	1	191	24
Deep	0	268	41

TABLE V
RESULTS OF PROCESSING TIME, OUTPUT MEAN AND STANDARD DEVIATION FOR VIDEO 2.

Network	Time (s)	Output mean	Standard Deviation
mini-VGG	0,25	0,949	0,021
Fast	0,10	0,969	0,027
Deep	0,15	0,971	0,026

second, which is relevant for online applications. The Softmax values obtained for both are higher than mini-VGG scores, demonstrating a robust identification of the pole.

With respect to the lost poles and false positives, shown in Tables II and IV, the superiority of the proposed networks is clear, with the deep topology reaching 23% more frames with recognized poles in video 1 and 40% more in video 2, which assists identifying failures and different problems with them. On the other hand, the number of lost poles for both proposed topologies is the same, favoring the fast network with lower computational cost.

The results and comparison can also be seen in the YouTube videos, at the links https://youtu.be/d9qV2A0cYfo for the deep network, and https://youtu.be/HqSE7WvH_MY for the fast.

IV. Conclusion

This paper presents a solution for the automatic recognition of poles around a railway using convolutional neural networks technique, in order to assist power grid monitoring of railways. In this work, two architectures are proposed and the comparison of results in practical situations were demonstrated.

Both networks presented accuracy rates higher than 93% for positives and negatives during training step. Even though the deep topology has shown superiority in poles found, the mean execution time of the fast one makes it the best choice for online applications.

It is important to emphasize that, with the used technique and relatively small amount of images used in training, when compared with the extension of the railway, the pole's features were acceptably identified, as demonstrated in Section III, even with a variety of backgrounds in different frames, which leads to a favorable result in various parts of the railway.

As a future proposition, other elements of interest from the railway can be included, mostly related to the previously identified poles, such as power transformers.

After that, different kind of machine learning techniques can also be applied, such as optimization methods proposed in [7], [18], [19], [20], [21].

Other important works about optimization techniques to be followed are [22], [23], [24].

Acknowledgment

The authors thank INERGE, UFJF, CEFET-MG, CPFL, TBE and the Brazilian company MRS-Logística for the financial support.

References

[1] J. Chen, Z. Liu, H. Wang, and K. Liu, "High-speed railway catenary components detection using the cascaded convolutional neural networks," in *International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6, IEEE, 2017.

[2] A. d. S. Lima Filho, "Manutenção em redes de distribuição de energia elétrica," *Repositórios de relatórios - Engenharia Elétrica*, no. 1, 2015.

[3] J. Chen, Z. Liu, H. Wang, A. Núñez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 2, pp. 257–269, 2018.

[4] C. Lueckmann, "Inspeção em linhas de distribuição de energia elétrica lages," *Repositórios de relatórios - Engenharia Elétrica*, no. 1, 2015.

[5] V. F. Vidal, L. M. Honório, M. F. Santos, M. F. Silva, A. S. Cerqueira, and E. J. Oliveira, "UAV vision aided positioning system for location and landing," in *18th International Carpathian Control Conference (ICCC)*, pp. 228–233, IEEE, 2017.

[6] M. F. Silva, A. S. Cerqueira, V. F. Vidal, L. M. Honório, M. F. Santos, and E. J. Oliveira, "Landing area recognition by image applied to an autonomous control landing of VTOL aircraft," in *18th International Carpathian Control Conference (ICCC)*, pp. 240–245, IEEE, 2017.

[7] L. M. Honório, M. Vidigal, and L. E. Souza, "Dynamic polymorphic agents scheduling and execution using artificial immune systems," in *International Conference on Artificial Immune Systems*, pp. 166–175, Springer, 2008.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[9] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2018.

[10] M. S. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwickit, "Transforming sensor data to the image domain for deep learning - an application to footstep detection," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 2665–2672, IEEE, 2017.

[11] C. Silva, D. Welfer, F. P. Gioda, and C. Dornelles, "Cattle brand recognition using convolutional neural network and support vector machines," *IEEE Latin America Transactions*, vol. 15, no. 2, pp. 310–316, 2017.

[12] A. Rosebrock, *Deep Learning for Computer Vision with Python*. PYImageSearch, 2017.

[13] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 2584–2589, IEEE, 2016.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] J. Masci, A. Giusti, D. Ciresan, G. Fricout, and J. Schmidhuber, "A fast learning algorithm for image segmentation with max-pooling convolutional networks," in *20th International Conference on Image Processing ICIP*, pp. 2713–2717, IEEE, 2013.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[18] L. M. Honório, A. M. L. da Silva, D. A. Barbosa, and L. F. N. Delboni, "Solving optimal power flow problems using a probabilistic α-constrained evolutionary approach," *IET generation, transmission & distribution*, vol. 4, no. 6, pp. 674–682, 2010.

[19] L. M. Honório, D. A. Barbosa, E. J. Oliveira, P. A. N. Garcia, and M. F. Santos, "A multiple kernel classification approach based on a quadratic successive geometric segmentation methodology with a fault diagnosis case," *ISA transactions*, 2018.

[20] E. J. Oliveira, L. M. Honório, A. H. Anzai, L. W. Oliveira, and E. B. Costa, "Optimal transient droop compensator and PID tuning for load frequency control in hydro power systems," *International Journal of Electrical Power & Energy Systems*, vol. 68, pp. 345–355, 2015.

[21] L. M. Honório, E. B. Costa, E. J. Oliveira, D. de Almeida Fernandes, and A. P. G. M. Moreira, "Persistently-exciting signal generation for optimal parameter estimation of constrained nonlinear dynamical systems," *ISA transactions*, 2018.

[22] L. C. Gonçalves, M. F. Santos, R. J. F. de Sá, J. L. da Silva, H. B. Rezende, *et al.*, "Development of a PI controller through an ant colony optimization algorithm applied to a SMAR® didactic level plant," in *19th International Carpathian Control Conference (ICCC)*, IEEE, 2018.

[23] F. F. Panoeiro, M. F. Santos, D. C. Silva, J. L. Silva, and M. J. Carmo, "PI controller tuned by bee swarm for level control systems," in *19th International Carpathian Control Conference (ICCC)*, IEEE, 2018.

[24] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, "Comparison of PID controller tuning methods: analytical/classical techniques versus optimization algorithms," in *18th International Carpathian Control Conference (ICCC)*, pp. 533–538, IEEE, 2017.