Towards Fire Identification Model in Satellite Images Using HPC Embedded Systems and AI

Jhon Deivy Perez Arguello[1,2,4(✉)], Carlos J. Barrios Hernández[1,2,4], and Julián Rodriguez Ferreira[1,2,3,4]

[1] Universidad Industrial de Santander (UIS), Bucaramanga, Colombia
jhon2198570@correo.uis.edu.co, {cbarrios,jgrodrif}@uis.edu.co
[2] High Performance and Scientific Computing Center (SC3UIS), Bucaramanga, Colombia
[3] Electronic Control, Simulation and Modelling Group (CEMOS), Bucaramanga, Colombia
[4] Advanced and Large Scale Computing Group (CAGE), Bucaramanga, Colombia

Abstract. Forest fires and environmental disasters that are rarely avoided due to Forest fires are environmental disasters is a crucial problem to resolve with High Performance Computing (HPC) due to the real-time need to avoid the reaction of the control agencies and the community. One of the strategies to support early warnings related to forest fires is using space technology and realtime image treatment. However, the large amount of data given by the satellite images, the cost of the satellite technology, and the difficulty of accessing remote places information make it challenging to deal with the problem. This project presents the development of a solution that fights fires through identification supported using artificial intelligence (AI), mainly Convolutional Neural Networks (CNN) and Computer Vision (CV). Space technology captures images in various spectral frequency ranges by optical instruments onboard artificial satellites. In addition, the solution deploys on a low-cost and easily accessible open-source embedded system, which allows its scope to be extended for use on mobile device applications such as robots, and uncrewed aerial vehicles, among others. This paper reflects the progress achieved within the project, mainly the creation of an open-source dataset of satellite images for fire classification, the election, conditioning, and training of the CNN.

Keywords: Satellite images · Computer vision · HPC embedded system · Artificial intelligence · Data analytics · Convolutional neural networks · Open-source

1 Introduction

The exponential development of the hardware required to execute algorithms and computational techniques has enabled the use of technologies such as Computer Vision (CV) or Machine Learning (ML) for solving specific problems [1]. These problems are notable for their intensive data handling in processes with high computational costs. However, advances in micro and nano electronics promote the use of compact (embedded) devices with low monetary cost and energy consumption as support for the solution to these

P. Navaux et al. (Eds.): CARLA 2022, CCIS 1660, pp. 103–115, 2022.
https://doi.org/10.1007/978-3-031-23821-5_8

problems [2]. One of these embedded systems, the NVIDIA Jetson Nano card [3], is widely used in prototyping and academic projects due to its good quality/price ratio.

One of these problems is finding fires from orbit. Although there are tools for this task [4], its identification continues to develop due to continuous improvements in algorithms, image processing, available instruments, and satellite communication.

Free access images from various satellites that observe the Earth at different wavelengths are used during this research. A dataset was created in which forest fires were observed from when they were small conflagrations until they became large fires. Finally, various characteristics are identified in the images, which using CNN and Computer Vision algorithms in embedded systems, allow the identification of said fires.

This paper presents the progress achieved in creating an alternative fire classification model to the current solutions, as an early warning so that the pertinent organisms act and evaluate the damage. For this, in the next Related Works section, a state of the art and some similar projects are exhibited resalting application goals, in this case, satellite imagery. The results section shows the creation of a dataset using images from the VIIRS sensor of the NOAA-20 and S-NPP satellites in which forest fires were analyzed from when they were small until they became large conflagrations, and later the performance of the CNNs in training with these images is shown. Finally, some conclusions and further work are presented.

2 Related Works

This section shows, firstly, a project that proposes a workflow for the identifying objects in satellite images with the help of machine learning technology, and secondly, a project to create small CNNs that reduce the computational load, useful to be deployed on embedded systems.

2.1 Satellite Imagery Multiscale Rapid Detection With Windowed Networks

Detecting small objects over large areas is a significant challenge in satellite imagery analysis. The main problems they face are a large number of pixels (more than 250 million), the geographical extension (more than 64 Km^2), or the tiny size of the objects of interest (less than 10 pixels). For which this research proposes a workflow called Satellite Imagery Multiscale Rapid Detection with Windowed Networks (SIMRDWN) [5], which evaluates satellite images of an arbitrarily large size at native resolution at speed greater than or equal to 0.2 km^2/s.

The SIMRDWN pipeline includes and compares the performance of some frameworks, where a version known as YOLT [6] is found, along with the TensorFlow object detection API models: SSD [7] and FASTER R-CNN [8]. This allows objects of very different scales to be quickly detected with relatively little training data across multiple sensors.

2.2 Lapped Convolutional Neural Networks for Embedded Systems

CNN has made numerous advances in many artificial intelligence applications. However, its complexity is quite relatively, usually requiring an expensive GPU (Graphics Processing Unit) [9] or FPGA (Field Programmable Logic Gate Array) [10] implementation, which is not cost-effective for many embedded systems. In this project, a new CNN or Lapped CNN (LCNN) architecture is developed that is suitable for resource-limited embedded systems [11].

The network is designed so that it can be decomposed into two or more stages. A hardware module can implement each with low complexity and low-resolution input.

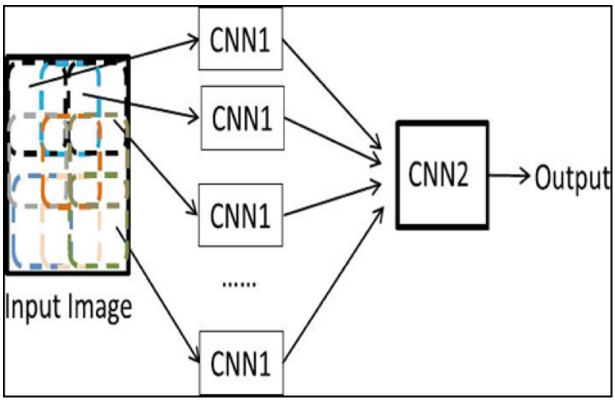

Fig. 1. Lapped convolutional neural networks.

Figure 1 show that the original input image is divide into some sub-images of the same size, with correctly designed overlaps with each other. The hardware module that implements the first stage of the CNN processes these sub-images sequentially. The outputs of different sub-images are merge and processed in the following low-cost hardware CNN module.

The result is the same as applying a larger-scale CNN to the entire image with higher resolution. Therefore, a lower-cost, larger-scale CNN system can be achieve by reusing inexpensive hardware CNN modules. Experimental results demonstrate the performance of the proposed scheme. This approach enables more cost-effective CNN solutions for some embedded systems. It is well suited for applications where basic deep learning capabilities are required but where constraints on computational cost and power consumption must also be met.

This proposal is in the convergence of the recognition of objects in satellite images with the deployment of CNNs in embedded systems. It seeks to present another solution to the identification of forest fires that contains the union of said technologies following an implementation methodology shown in the following section.

3 Workflow

The methodology used in this research work is based on the AI life cycle [12], grouping 19 development stages into 4 phases as seen in Fig. 2.

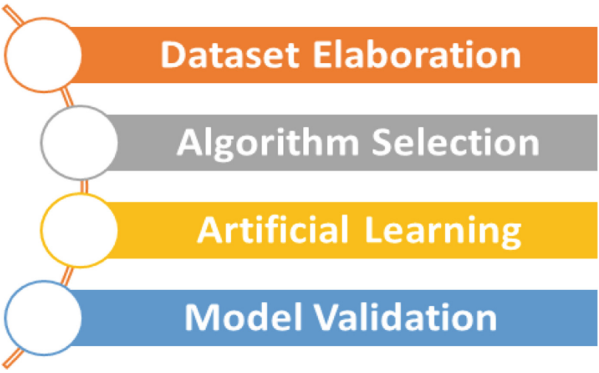

Fig. 2. Project development methodology.

Figure 2 shows the methodology implemented in this research project. It begins with the elaboration of a dataset of images acquired from a public platform. It continues with the selection of the most relevant object classification deep learning algorithm to implement on the JETSON Nano embedded development board. The next stage is the training of the neural network with the structured data set. The last activity corresponds to the validation of the model through its deployment in the embedded system.

3.1 Dataset Elaboration

The satellite images that make up the dataset were obtained from NASA's public access platform called WORLDVIEW [13], which shows photographs of the eEarthin a timeline with different wavelengths. From there, 1100 images were downloaded in various combinations of spectral bands generated by an optical sensor called VIIRS [14], present in satellites of the National Oceanic and Atmospheric Administration-20 (NOAA-20) [15] and the National Program of Association in Suomi Polar Orbit (S-NPP) [16]. These combinations of spectral bands are identified as "M3-I3-M11" [17], and their characteristics can be observed by analyzing the bands captured by the VIIRS sensor.

We chose the satellite images with this spectral combination due to the high contrast generated by the color of the thermal radiation of the fire, concerning the rest of the image and of the smoke given off by the fire for the clouds and other aerosols, allowing a more efficient identification with the algorithms used in this project (Fig. 3).

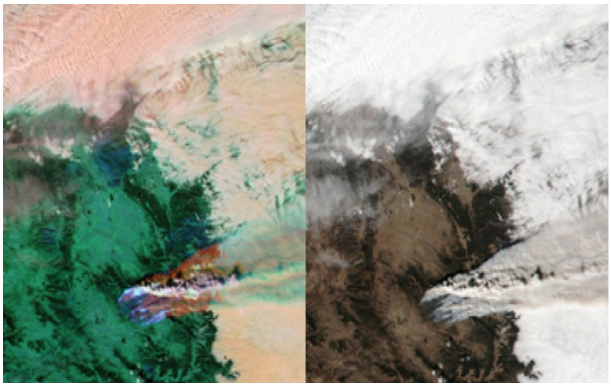

Fig. 3. Bands M3-I3-M11 left. Bands True Color, right.

The 1,100 images were divided into two packages. The first of 1000 images are used for the neural network training process, 500 with forest fires and 500 without fires. The second of 100 images are used for the classification process with the algorithms, 50 with forest fires and 50 without fires. The characteristics of these images are as follows:

- Training images size: 224 × 224 pixels
- Identification images size: 2048 × 2048 pixels
- Spatial resolution: 125 m
- Altitude: 200 km (Average altitude for microsatellites)

For the training images, 80% (800) were designated for training and 20% (200) for validation. Due to the low number of images and as a recommended practice in convolutional neural network training, a technique is applied to the first set of images (800) that allows us to increase our training data set caled "Data Augmentation" [18]. With this technique, we apply transformations to the original images, generating others with the reflected changes. The modifications applied in this case are Flip and Rotation. As a result, the 800-image set now has 2,230 images, and adding this with the 200-image validation set, leaves 2,430 images for the entire neural network training process.

3.2 Algorithm Selection

Two CNN models were chosen for training and evaluation to identify forest fires in satellite images: Inception V3 [19] and Mobilenet V2 [20]. These models are selected for their affinity with the project purposes, which implicitly entails generating a minimum expenditure of resources without losing effectiveness in the classification process.

The first model is inception V3 [21], which focuses mainly on consuming less energy and computational resources by modifying the previous models. We find factored convolutions, regularization, dimension reduction, and parallel calculations within the techniques used to optimize performance.

On the other hand, we have the mobilenet V2 model [22], designed for deployment on mobile devices; therefore, one of its characteristics is the low computational and energy consumption required for its operation. This model retains many of the features of its predecessor model, mobilenet V1 but introduces two new advances, such as linear bottlenecks between layers and shortcut connections between jams.

Regardless of the neural network model chosen to perform an classification process, good training performance requires a large number of images. Due to the difficulty of construction and the time needed to consolidate a large dataset, it is necessary to rely on a specific technique for these situations called transfer learning [23]. This technique take advantage of the knowledge acquired from previously trained models to train new models, which do not necessarily need to have a large number of images. For this project, it is necessary to carry out transfer learning using the pre-trained weights in the two neural models of the ImageNET image dataset [24].

The Inception V3 model used in this project is adapted from a development that detects fire in images captured by indoor and outdoor security cameras. On the other hand, the Mobilenet V2 model is adapted from a tutorial created by Tensorflow developers [25] and modified in the Roboflow platform [26] for flower classification. Data of this experiment are available for reproducibility in the project's GitHub repository [27].

4 Results

Now, the results obtained in the third methodological phase are presented below. In addition, the evaluation metrics that will be used are shown, although the model has not been validated on the embedded system.

4.1 Artificial Learning

This process begins with loading the structured dataset corresponding to the first methodological stage. The pre-trained CNN is also loaded, which works as a feature extractor by stacking the upper classification layers. Then the model is trained with our dataset and the parameters are saved as an "h5" file that will be used as input in the inference algorithm. The training results for each model were as follows:

In Fig. 4, the Inception V3 model in training shows an increase from approximately 83% to over 95% accuracy after five (5) epochs and oscillating between 94 and 96% in the remaining fifteen (15) epochs. The validation stage, the level of precision is between 90 and 93%.

Figure 5 shows that the inception V3 model during training presents a descending level of loss in a staggered manner from approximately 80% to 10% after eight (8) epochs, then stay there. In the validation stage, the level of loss always fluctuates between 20 and 30%.

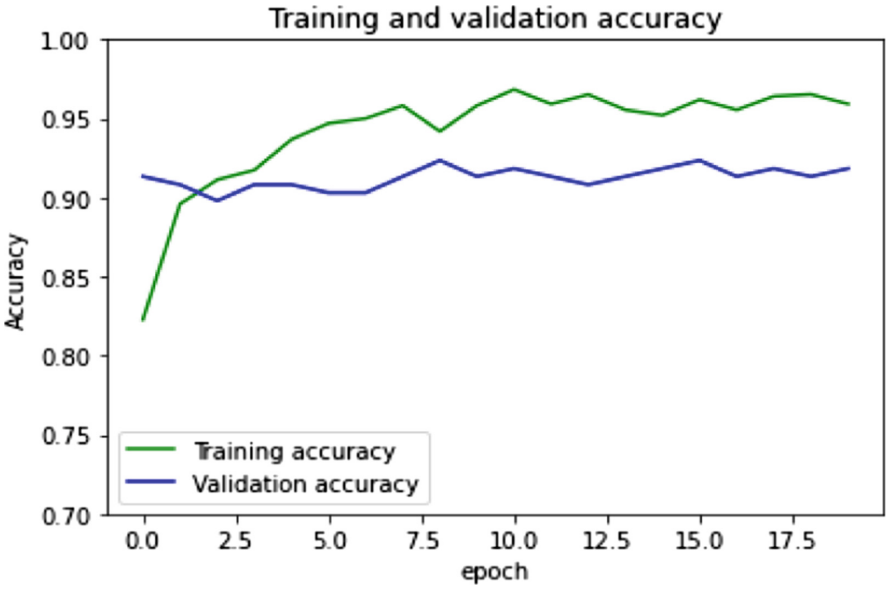

Fig. 4. Accuracy inception.

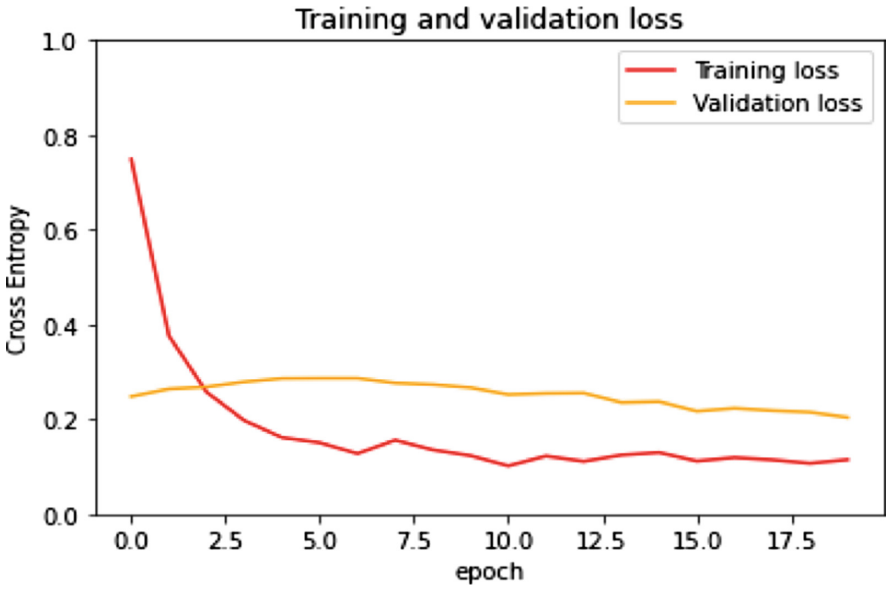

Fig. 5. Loss inception.

As can be seen in Fig. 6, the mobilenet V2 model in training presents a rectilinear ascent from approximately 87% to 96% accuracy after two (2) epochs, and then continues with a shallow ascending parabola until reaching 99% during the four (4) and following

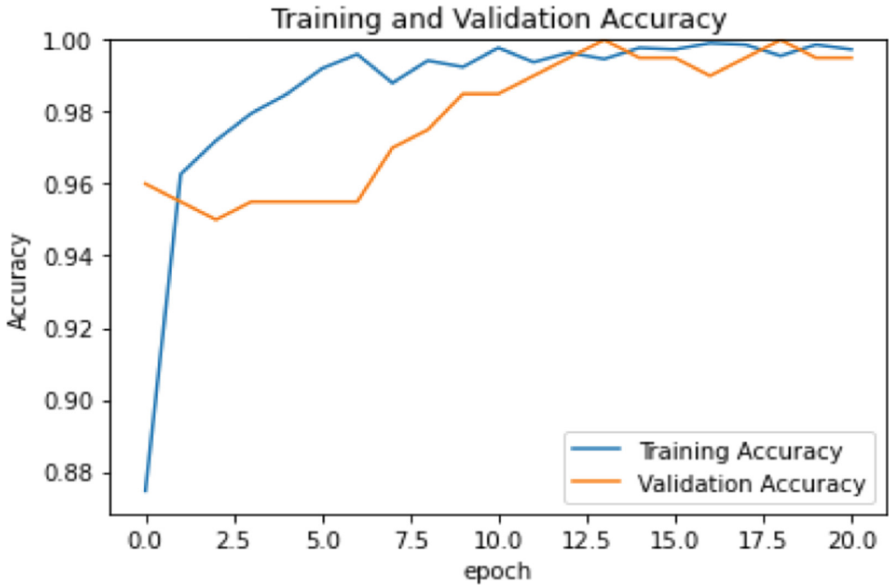

Fig. 6. Accuracy mobilenet.

fourteen (14) epochs. In the validation stage, the precision level also draws remains at 95% for six (6) epochs, and after an ascending parabola until reaching approximately 99%.

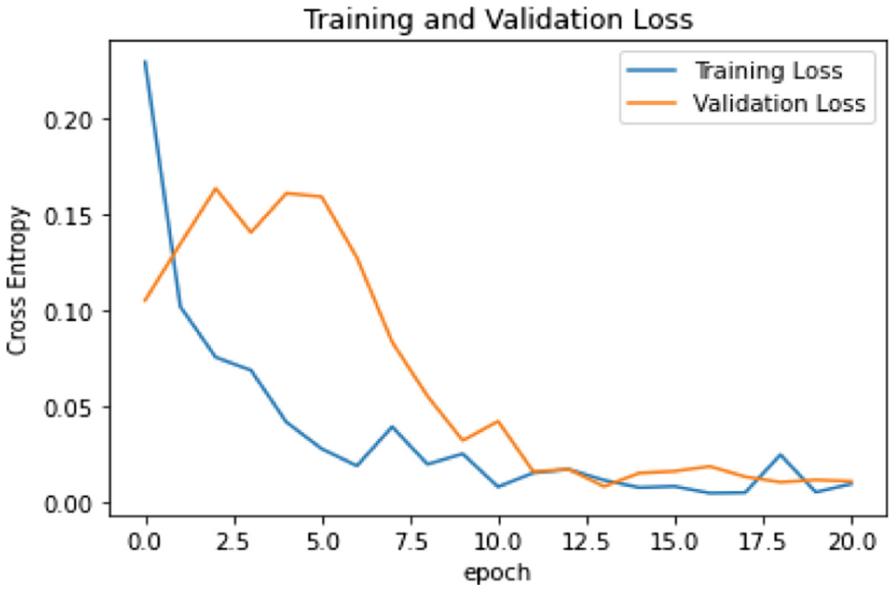

Fig. 7. Loss mobilenet.

Figure 7 shows that the mobilenet V2 model during training presents a level of loss through a descending curve from approximately 23% to 3% during six (6) epochs, to later mark a fluctuating line between 2 and 4%. In the validation stage, the level of loss draws a curve starting at 10%, rising to 16% for five (5) epochs, then falling and staying at 3% approximately.

For these two models, it is observed that the beginning of the precision is high because the training is supported by the characteristics extracted from the pre-trained model with the previous weights thanks to transfer learning. The same happens when we analyze the loss, although in the mobilenet model the training and validation start with higher percentages than in the inception model.

It is also observed that the increase in epochs in training and validation allows us to see more continuous values in the precision and loss metrics, as reflected in the curve of the mobilenet model, contrary to what is shown in the inception model.

4.2 Evaluation Metrics

4.2.1 Reliability

Reliability within the evaluation process allows determining the consistency level of the model through consecutive measurement events. For this, the Confusion Matrix will be used, which shows us the crossing of the true results with those obtained after the execution of the algorithm. Making use of the aforementioned prediction lists, the confusion matrix is created for each algorithm.

The results thrown by the confusion matrix are divided into 4 variables that reflect what happens when comparing the expected or real results with those received or generated by the algorithms (see Table 1).

Table 1. Confusion matrix result.

Real/Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Where:

- TP - true positive: is real positive and predicted positive.
- FN - false negative: is real positive and predicted negative.
- FP - false positive: is real negative and predicted positive.
- TN - true negative: is real negative and predicted negative.

Based on these variables, the metrics that will allow evaluating the reliability of the algorithm are structured.

Precision: It is the resulting index between the true positives and the total positives generated by the algorithm.

$$TP/(TP + FP)$$

Accuracy: It is the resulting index between the successes, both positive and negative, and the total results generated by the algorithm.

$$\mathrm{TP} + \mathrm{TN} \,/\, (\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN})$$

Recall: It is the resulting index between the true positives and the total of real positives.

$$\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$$

F1-Score: Allows comparing two metrics into one, sensitivity and precision, and is widely used when the classes within the dataset are unequal.

$$2 * (\mathrm{Recall} * \mathrm{Accuracy}) \,/\, (\mathrm{Recall} + \mathrm{Accuracy})$$

4.2.2 Efficiency

The different ML techniques require a large amount of computational and energy resources for their deployment, for this reason, in the development of this type of model, it is important to evaluate the consumption of said resources and, if possible, reduce it. This approach is called Algorithmic Efficiency. Although it is true that in any situation the saving of resources is a matter of great attention, in this particular case it is even more so, due to the limitations presented by embedded systems or integrated architectures.

The metrics to consider with the project models are:

- Energy Consumption: Measured in milliwatts (mW) and corresponds to the average consumption of the embedded card during the test with each algorithm.
- RAM consumption: Measured in megabytes (MB). The Jetson Nano card shares RAM with the GPU, that is, it is not separated.
- Device Temperature: Measured in degrees centigrade (°C) and corresponds to the temperature recorded by the entire card during the elapsed time.
- GPU consumption: Measured in percentage (%) and corresponds to the use of the GPU in its processing.

5 Conclusion

The results obtained in the progress of this research show that with a small dataset, adequate training of a convolutional neural network can be carried out, allowing an alternative proposal to be proposed as support in the surveillance of forest fires.

In addition, a dataset of 2430 satellite images with spectral bands M3-I3-M11 was structured, captured, and conditioned from within the WORLDVIEW platform. This dataset is available to the academic or scientific community in the respective repository.

The training through transfer learning of the CNNs used in this project facilitated the implementation process of the development, due to the decrease in time and data

that these networks needed in their preparation compared to those carried out right from the start.

However, preparing an image dataset and building a model with convolutional neural networks presents several challenges for researchers, for example, the time spent developing prototypes that allow evaluating the behavior of a model, the reuse of these prototypes in other applications, and the adaptation of the resulting models to run on hardware with more limited specifications.

6 Future Work

Different proposals follow the outcomes of this project. First, the customization of the method to develop several fire classifications models, searching low-cost implementation, accuracy, and computing efficiency. Second, the growth of open access data sets to help researchers and agencies in the fire early aware forecasting. Finally, this project is under development and implementation for real use inside the Space Mission as A Service Model or SMMAS, an HPC-cloud computing model to space projects developed by the High Performance and Scientific Computing Center of the Universidad Industrial de Santander (SC3UIS),[1] and some of the following steps to be executed are:

- Execution Execution of the CNNs using minimal resources on the embedded system for inference with the images acquired.
- Measurement of the resources consumed during the execution process of the models on the embedded system to systems performance evaluation and characterization of the resources.
- Evaluation of the implemented models in the classification by a minimum expenditure of resources.

References

1. Feng, X., Jiang, Y., Yang, X., Du, M., Li, X.: Computer vision algorithms and hardware implementations: a survey. Integration **69**, 309–320 (2019)
2. Brozek, T., ed.: Micro-and nanoelectronics: emerging device challenges and solutions. CRC Press (2014)
3. Nano, N.J.: https://nvidia.com/en-us/autonomus-machines/embedded-systems/jetson-nano/. Accessed 23 Oct 2021
4. Petrescu, R.V., et al.: NASA satellites help us to quickly detect forest fires. Am. J. Eng. Appl. Sci. **11**(1), 288–296 (2018)
5. Van Etten, A.: Satellite imagery multi-scale rapid detection with windowed networks. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 735–743. IEEE (2019)
6. Van Etten, A.: You only look twice. Rapid multi-scale object detection in satellite imagery. arXiv preprint arXiv:1805.09512 (2018)

[1] More information in: https://www.sc3.uis.edu.co.

7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
8. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. CoRR abs/1506.01497 (2015). http://arxiv.org/abs/1506.01497
9. Graphic Processing Unit (GPU). https://www.intel.la/content/www/xl/es/products/docs/processors/what-is-a-gpu.html. Accessed 13 July 2020
10. Field Programmable Gate Array (FPGA). https://www.arm.com/glossary/fpga. Accessed 14 July 2020
11. Wang, X., Ng, H.W., Liang, J.: Lapped convolutional neural networks for embedded systems. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 1135–1139. IEEE (2017)
12. De Silva, D., Alahakoon, D.: An artificial intelligence life cycle: From conception to production. Patterns, 100489 (2022)
13. National Aeronautics and Space Administration, data visualization application WORLDVIEW. https://worldview.earthdata.nasa.gov/. Accessed 20 Oct 2020
14. Cao, C., Xiong, J., Blonski, S., Liu, Q., Uprety, S., Shao, X., Weng, F.: Suomi NPP VIIRS sensor data record verification, validation, and long-term performance monitoring. J. Geophys. Res. Atmos. **118**(20), 664–678 (2013)
15. Cao, C., et al.: NOAA-20 VIIRS on-orbit performance, data quality, and operational Cal/Val support. In: Earth Observing Missions and Sensors: Development, Implementation, and Characterization V, vol. 10781, p. 107810K. International Society for Optics and Photonics (2018)
16. Weng, F.: Advanced Technology Microwave Sounder Calibration and Validation. Liang, S.: Comprehensive Remote Sensing, pp. 42–63. Elsevier (2018). ISBN 9780128032213. https://doi.org/10.1016/B978-0-12-409548-9.10393-8
17. National Oceanic and Atmospheric Administration. NOAA Technical Report NESDIS 142 (2017). https://ncc.nesdis.noaa.gov/documents/documentation/viirs-users-guide-tech-report-142a-v1.3.pdf. https://doi.org/10.1016/B978-0-12-409548-9.10393-8. Accessed 15 Apr 2020
18. Zoph, B., Cubuk, E., Ghiasi, G., Lin, T., Shlens, J., Le, Q.: Learning data augmentation strategies for object detection. arXiv (2019)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016)
20. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
21. D. Shah, «Early Fire detection system using deep learning and OpenCV,» 2020, https://towardsdatascience.com/early-fire-detection-system-using-deep-learning-and-opencv-6cb60260d54a. Accessed 20 Jun 2021
22. Sahoo, S.: How to Train MobileNetV2 On a Custom Dataset (2021). https://blog.roboflow.com/how-to-train-mobilenetv2-on-a-custom-dataset/. Accessed 24 Jun 2021
23. Sarkar, D., Bali, R., Ghosh, T.: Hands-on Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. Packt Publishing (2018)
24. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)

25. Pang, B., Nijkamp, E., Wu, Y.N.: Deep learning with tensorflow: a review. Journal of Educational and Behavioral Statistics **45**(2), 227–248 (2020)
26. Roboflow Platform. https://roboflow.com/. Access 30 Oct 2021
27. Github. https://github.com/jhonesis/Proyecto-IGNIS. Access 13 Jun 2022