Performance Analysis of Machine Learning-Based Systems for Detecting Deforestation

Michel A. L. de Araújo[1], Ermeson Andrade[1] and Fumio Machida[2]

***Abstract*— Remote sensing has become an important tool to recognize the land and objects on the ground by analyzing sensor data. The use of Machine Learning (ML) algorithms for image classification has increased in recent years. ML-based image classifiers for remote sensing play an important role in detecting deforestation, illegal mining or fire. However, the precise classification of land use is a challenge, especially in remote tropical regions, due to the complex biophysical environment and the limitations of the remote sensing infrastructure. This work aims at studying the trade-offs between performance and accuracy of classification systems for the Brazilian Amazon rainforest, taking into account different computing plataforms (server and edge), ML algorithms and qualities of images. Although there is a direct relationship between accuracy and the quality of the images, our experimental study shows that it is still possible to use low-cost computer environments to perform the image classification with a comparative accuracy. The result indicates that Amazon rainforest can be monitored with affordable computing resources such as edge computer on a drone.**

I. INTRODUCTION

The use of remote monitoring (RM) has become crucial for various application areas, such as urban recognition and environmental protection. The Amazon rainforest, for example, which is a symbol of biodiversity and also known as the lungs of the world, depends a lot on remote sensing (RS) due to its vast territory. Since the 1990s, the Brazilian government's INPE (National Institute of Space Research) together with international research institutions, such as United States Geological Survey (USGS) and National Aeronautics and Space Administration (NASA), have developed remote sensing policies in the Amazon rainforest aiming to map the use of this vast territory [1]. The use include housing, family farming, legal and illegal mining and deforestation. Several algorithms, specifically those in the Computer Vision (CV) and Pattern Recognition (PR) fields, have been successfully developed to classify satellites images such as Landsat 8 OLI (Operational and Land Imager) and MODIS (Moderate Resolution Imaging Spectroradiometer).

Generally, the RS is carried out as follows. Geospatial satellites take pictures of areas of interest over a period of time. Then, the images are sent to remote servers, so that specialized professionals analyze the images in search of changes, such as new deforestation, fires, illegal mining, among others [2]. This approach can be very time-consuming because it requires a lot of manual work and attention from the professionals. Over the years, several approaches have been proposed for remote sensing. For instance, Maretto [2] proposed an autonomous process for detecting land use and land cover through Deep Learning (DL) algorithms, such as CNN and its variations. This work represents a great advance for the environmental protection, since images can be analyzed automatically by computers with almost human-like accuracy. Nevertheless, a major disadvantage of this approach is the size of the pictures taken by satellites, which are typically very large[3]. Consequently, it requires servers with high computational power to process DL algorithms that causes an expensive solution

As Brazilian forests are vast, and the scale and intensity at which illegal deforestation occur grows every year [4], it is necessary to use tools capable of responding quickly to these situations. Drone-based surveillance systems have been used for that purpose [5]. With edge computing capabilities, drones are a viable alternative for executing operations to combat illegal activities in the Amazon basin. Additionally, drones have some advantages over traditional satellite monitoring system, such as (i) drones are much cheaper and more accessible, (ii) they are easy to use, (iii) drones can visit an area multiple times a day, and (iv) images taken by a drone do not suffer from the presence of clouds. However, the performance trade-offs of using DL algorithms for detecting deforestation in these edge-based environments needs to be investigated.

Several works have demonstrated the effectiveness of Deep Neural Networks (DNN) algorithms for Land Use and Land Cover (LULC) and LULC temporal change detection [6]. However, to the best of our knowledge, none of existing works has considered the performance differences of these algorithms on edge and server platforms. Thus, this work aims at studying the trade-offs between performance and accuracy of ML classification systems for the Brazilian Amazon rainforest. More specifically, we investigate experimentally the impacts of image sizes and types of algorithms on the accuracy and performance in the differente computation platform that is either an edge computer or a server. Our results reveal two important insights. First, the classification time on the edge environment is almost two times longer than the server environment due to its lower processing capacity. Specifically, for the CNN algorithm, where the classification time was of 0.125 seconds on the edge and 0.07 seconds on the server. For the RF and KNN algorithms, the difference is almost imperceptible. Second, even reducing the size of the images, it is still possible to obtain an average accuracy of

[1] Department of Computing, Federal Rural University of Pernambuco (UFRPE), Recife, PE, Brazil. michel.arruda@ufrpe.br, ermeson.andrade@ufrpe.br

[2] Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki, Japan. machida@cs.tsukuba.ac.jp

approximately 0.93 for the image size of 32x32 and 0.956 for the size of 128x128 at the CNN model. We hope this work can provide low-cost surveillance options that can be used in developing regions with little access to capital (e.g.: indigenous community).

The remainder of the paper is organized as follows. Section II presents the related work. Section III introduces fundamental concepts adopted in the paper. Section IV describes the proposed approach. Section V describes the experimental results. Finally, the Section VI presents the conclusions and briefly introduces future work.

II. Related works

The use of ML for detecting deforestation is a relatively new topic that has attracted the attention of researchers. In [7], a detailed survey on the different ML algorithms for detecting deforestation is presented. To position our paper and indicate its contributions, we first summarize some related work that has been done in the classification of satellite images. Next, we discuss the related work with respect to drone applications. Lastly, we provide a comparison of our work in relation to the existing works in terms of modeling and evaluation.

Several techniques for classification of satellite images have been developed in recent years. Zhang el atl. *et al.* [8] showed the advantages of using RM and how DL algorithms can be combined with satellite images to extract value. Besides, the authors provided a technical explanation of the Convolutional Neural Network (CNN), autoencoder, Restricted Boltzmann Machines (RBM), and sparse coding algorithms, in addition to illustrating all the basic processes of using DL techniques for RM. In [8], *Zhu et al.* [9] performed an extensive review on the use of DL in RM images, explaining a greater variety of algorithms and, explaining how different spectral bands are used to perform different classification tasks in RM images, such as target detection, semantic segmentation and pixel-wise classification.

Recently, drone-based systems have been used for image classification. In [10], Kovacs *et al.* presented a review of the use of systems based on unmanned aerial vehicles (UAVs) for activities related to safety, such as immediate response to disasters and public and civil security. Peneque-Gálvez *et al.* [5] conducted a feasibility study on the use of drones by communities to monitor local deforestation. The work showed the benefits not only for the community itself, but also for partner organizations and the general public.

Considering the works available in the literature, none of the studies aimed at evaluating the performance of server and edge environments in the context of deforestation. In contrast, this work aims to evaluate various machine learning algorithms developed for detecting deforestation in order to explore the *trade-offs* between performance and accuracy. As the computational resources required to run ML algorithms varies widely, we also investigate the performance impacts of these algorithms deployed in server and edge environments.

III. Background

A. Deforestation

Brazil has the largest tropical forest in the world with an area of approximately five million square kilometers. This forest is home to the richest biodiversity of any ecosystem on the planet. Nevertheless, the rate of deforestation in Brazilian tropical forests is among the highest globally. According to INPE, 2,254 square kilometres of tropical forest were removed from July 2018 to July 2019, an increase of 278% over one year [11]. Deforestation is becoming a global problem with extensive environmental and economic consequences, since it reduces biodiversity, impacts on climate change by CO2 emission, and breaks up indigenous communities [12].

B. Remote monitoring (RM)

RM is the science of acquiring, processing, and interpreting images and related data, obtained from aircraft, satellites or drones. It has been used for many decades. The advantages of RM include the ability to collect information over large spatial areas, to identify natural features or physical objects on the ground, to observe objects or areas on a systematic basis, etc [8]. In the context of deforestation, RS is essential. Satellite or drone-based systems can cover areas that are difficult to access and dangerous or environmentally sensitive areas. It can also identify where mining, new agriculture fields, or other deforestation have replaced native forest and serve as evidence for authorities to take actions against the violators. It is worth to highlight, however, that different from tradition satellite imagery, drones can obtain very high-resolution images from cloudiness areas and do not have to wait for satellites to pass over the monitoring area.

C. ML techniques for detecting deforestation

Machine Learning is a field of Artificial Intelligence. It consists of algorithms that allow the machine to learn from data, without being explicitly programmed. There are a good number of ML techniques for detecting deforestation, such as Support Vector Machine (SVM), CNN, Siamese Convolutional Network (S-CNN), among others. SVM is one of the most popular supervised machine learning algorithm used in image classification systems because it performs well when the data set has few labeled samples [13]. Random Forest methods [14] are also commonly used in image classification tasks. However, when it comes to accuracy, deep learning, which is a class of machine learning algorithms, is the choice. It can learn by employing several layers of neural networks, resulting in a much better overall accuracy. As DL algorithms are much more complex and typically require much more data to deliver better predictions, they require more computational resources to be trained and executed. Consequently, they might not be a good fit for edge-based enrolments like drones.

IV. Experimental Methods

A. Data Structure

In this paper, a dataset from a 2017 Kaggle competition called *Planet: Understanding the Amazon from Space* was used. The objective of this competition was to label around 40000 images from Brazilian rainforest and build algorithms to classify these images with respect to the land use. The dataset consists of 40479 jpg images with the size of 256x256 pixels. They are classified according to 17 different labels and each image can have more than one label - thus being a multi-label classification problem, as shown in Figures 1, 2 and Table I. For more information concerning the meaning of each of the labels, the reader should refer to [15]. The dataset was divided into a ratio of 80:10:10, where 80% was used for training, 10% for test and 10% for validation. Learning Scikit-Learn 0.20.0 and Tensorflow 2.2.0 were used for creating and executing the models.

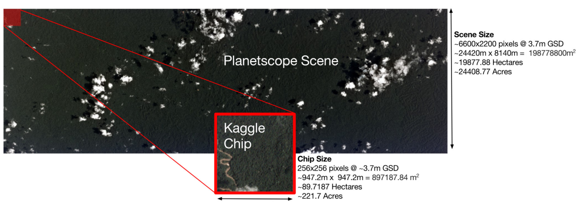

Fig. 1: Kaggle scene and chips

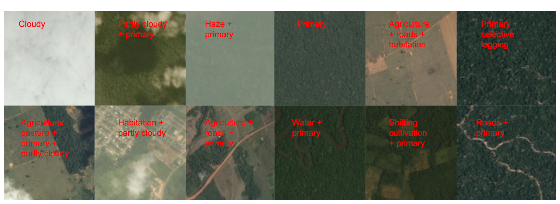

Fig. 2: Kaggle chips and possible labels for each image

TABLE I: Images names and its possible labels

image name	tags
train_0	haze primary
train_1	agriculture clear primary water
train_2	clear primary
train_3	clear primary
train_4	agriculture clear habitation primary road

B. Environments

For the experiments, we adopted two environments for the classification task: a server environment and an edge environment. As satellite images are, usually, sent to a remote server for processing, we adopted initially a server environment for the classification task. On the other hand, drone-based systems perform its classification task locally. Consequently, we adopted an edge computing platform for the drone using a Raspberry Pi model 4. The settings for the adopted environments are detailed in Table II.

TABLE II: Adopted environment.

Conf.	Server	Edge
CPU	Intel i5 4 Gen	Quad core Cortex-A72
RAM Memory	8Gb DDR3 1300Mhz	4Gb DDR3
Disk Storage	500Gb HD	16Gb SD card
GPU	Integrated	Integrated

C. ML algorithms

Three DL algorithms were adopted to perform the classification task, which included KNN, RF and CNN. The KNN, which is the most basic one, was trained using $K = 19$ (lowest error rate) for the number of closest neighbors. KNN is a peculiar algorithm for our problem in question because to classify a new image it loads the entire training set into memory and then performs the comparison one by one. Thus, we anticipate it might not not suitable for the edge environment, which has limited computational resources.

The second algorithm adopted was the RF. Despite the fact that the random forest do not have high capacity for abstraction of characteristics, it has been widely used for classification of remote sensing images. We conducted the training and classification by varying the number of trees (n_trees=100 and n_trees=500) in order to study the trade-offs between accuracy and performance.

The third algorithm adopted in our analysis is a well known algorithm in the field of computer vision called CNN. This algorithm has a high capacity for classify images. For CNN, we used 3 blocks with 2 convolution layers followed by a 2x2 Maxpooling. Not that, as previously mentioned, DL algorithms are generally highly complex and carry a lot of information due to their high level of feature abstraction. Consequently, the impact on the performance of this type of algorithm is important to be studied, especially for edge-based environments.

As edge environments are resource-constrained environment, all algorithms were trained considering different image sizes in order to analyze the trade-offs between image size and accuracy. Thus, the images were reduced from their original size to sizes of 128x128, 64x64, 32x32, 16x16 and finally 8x8. It is worth to highlight that the algorithms were trained on the server environment, but the classifications were performed on both server and edge environments.

D. Evaluation of the models

It is of great importance to choose the right assessment method for our models. Since our case is a multi-label classification problem, the coefficients adopted for the evaluation of the models were Precision, Recall, Accuracy and F1 score [16]. Each of these coefficients tell something about the quality of the algorithm. But, as we are dealing with land use classification with potentially dangerous situations (such as illegal mining or selective deforestation), it was chosen to prioritize the Recall coefficient because it provides the lowest possible number of false negatives. Note that, in the context of deforestation, authorities do not want deforestation activities to go unnoticed.

V. Results

This section presents results that cover both the usefulness of the adopted computing environments and a comparison of ML algorithms for predicting deforestation in terms of accuracy and performance.

A. *Server environment*

We start by describing the results of the training and tests carried out in the server environment. The first algorithm trained and tested was CNN. For this algorithm, the hyper-parameter threshold was defined, where its values range from 0.1 to 0.9. *Threshold* is the probability of the prediction being classed as positive or negative. For example, when setting the threshold equal to 0.4, if the algorithm returns that there is a 0.5 probability of existing a road in the image, it will be classified as positive, since 0.5 is greater than 0.4. Otherwise, it will classified as negative. The purpose of this hyperparameter is to adjust the accuracy of the model to the needs of the problem. Since it is very important that no illegal deforestation activity goes unnoticed, it is necessary to have a high Recall. Note, however, that increasing the threshold value, the Recall decreases. That is why it is necessity to set the parameters according to the problem. Nevertheless, for the deforestation problem, accuracy is not a priority, as false positives is less harmful than ignoring a true positive.

Figure 3 shows the impacts of the threshold on Accuracy. In this graph, the x-axis represents the values for threshold, while the y-axis represents the accuracy obtained by the model. As can be seen, the higher the threshold and image sizes, the better the accuracy. However, accuracy by itself may not be the best evaluation metric, specially for the deforestation case. Thus, it is necessary to analyze other metrics like Precision and Recall.

Figure 4 shows the impacts of the threshold on Precision and Recall. The left graph shows the relationship between threshold and Precision. The right graph shows the relationship between threshold and Recall. It is expected that, by increasing image sizes, the quality of the model would also increase. However, we noticed that the increase in Precision and Recall is almost imperceptible when the image size goes from 64x64 to 128x128. The results also show that the values of the two coefficients are inversely proportional to each other and thus understanding their differences is important in building an efficient classification system. Therefore, the value chosen for the threshold is 0.3, as it provides the most balanced values of Precision and Recall. Table IV summarizes the results for the CNN assessment considering the threshold equal to 0.3.

TABLE III: Evaluation Results for CNN model

Image size	Precision	Recall	Accuracy	F1 Score
8x8	0.69	0.791	0.905	0.737
16x16	0.744	0.872	0.928	0.803
32x32	0.76	0.898	0.935	0.823
64x64	0.815	0.907	0.95	0.859
128x128	0.847	0.904	0.956	0.875

Figure 5 shows the time that CNN model takes to classify a single image in the server environment. The x-axis represents the size of the image, while the y-axis represents the average time (in seconds) that the model takes to classify a single image. As expected, the results show that the larger the image size is, the greater is its dimensionality and, therefore, the longer the model takes to process all the information. Note that the ratio between the longest and the shortest time is 1.57[s].

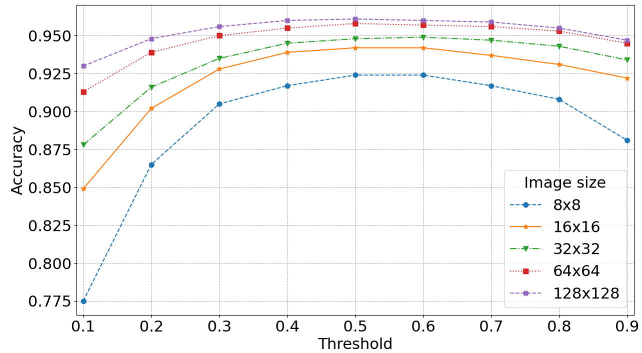

Fig. 3: Accuracy as function of Threshold for CNN.

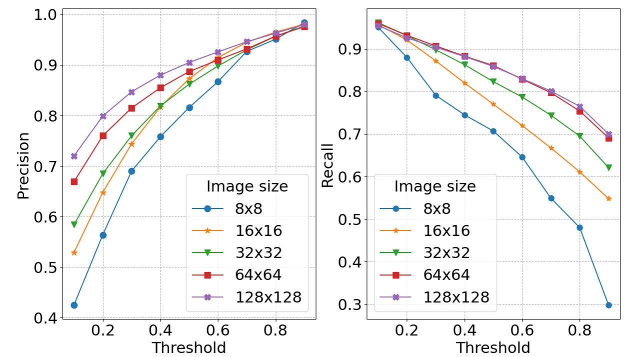

Fig. 4: Precision and recall as function of Threshold for CNN.

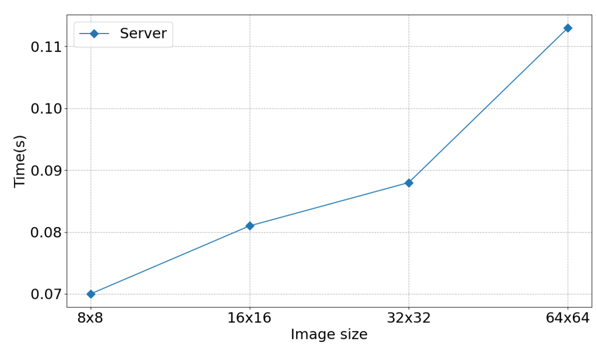

Fig. 5: Time to classify a single image for CNN in a server environment.

The next algorithm trained and tested was Random Forest. For this algorithm, the hyperparameter n_trees was used, which represents the number of estimators that the model should use in the training. The greater the number of trees, the greater the complexity of the model, and therefore, the heavier it becomes. Figure 6 show the results of the training for the RF. The left graph shows the relationship between Precision and number of trees, while the right graph shows the relationship between Recall and number of trees. The results reveal that the image size does not seem to have a linear influence on the coefficients, since the size of the image with the best Recall and Precision was 32x32.

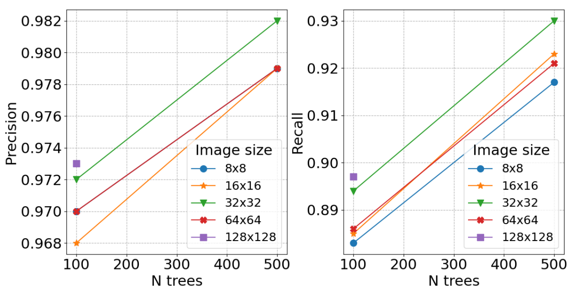

Fig. 6: Precision and Recall as function of the number of trees for the Random Forest algorithm.

Figure 7 shows the time to classify a single image in the server environment concerning the RF algorithm. In this graph, the x-axis represents image sizes and the y-axis represents the classification time for a single image. As expected, the model takes longer to classify images with $n\_trees = 500$ because it is more complex. The ratio between the longest and the shortest classification time based is 6.02. However, the server environment was unable to train the RF model for $n\_trees = 500$ and image size of 128x128, due to the lack of memory. Additionally, the results show that the sizes of the images do not have much influence on the classification time.

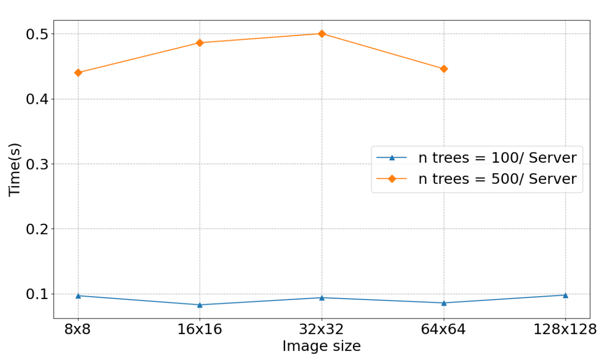

Fig. 7: Time to classify a single image for RF in a server environment.

Finally, we present the results for the KNN algorithm. The only hyperparameter used was the n_neighbors, so that we performed the training of the model for values of n_neighbors ranging from 1 to 40, and the value $n\_neighbors = 19$ was chosen as it resulted in the best possible result during training. The results show that the accuracy increases with the increase in the size of the images. However, Recall does not have a monotonic relationship with the size of the image. It is worth to note that the variation for the Precision between the highest and lowest value is only 0.1, while for Recall this variation is 0.2. Therefore, the best performance for KNN was using the image size of 64x64. Additionally, the server environment was unable to train the model for images of 128x128 due to lack of RAM. This show that although CNN models are more complex due to their deep layers of feature extraction, they are much more optimized than the KNN and RF models when it comes to computational performance.

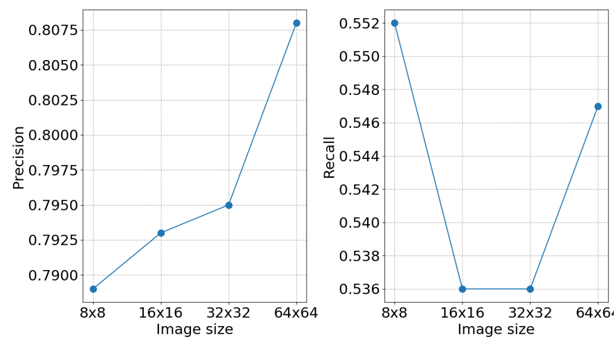

Fig. 8: Precision and Recall as function of image size for the K-Nearest Neighbors algorithm.

Figure 9 shows the result of the classification tests for the KNN. The results clearly show that as the image sizes increase, the classification time also increases. The ratio between the highest and the lowest classification time for the KNN is 47.7 (s). It means that it will be necessary to reduce the image size in order to increase the edge performance.

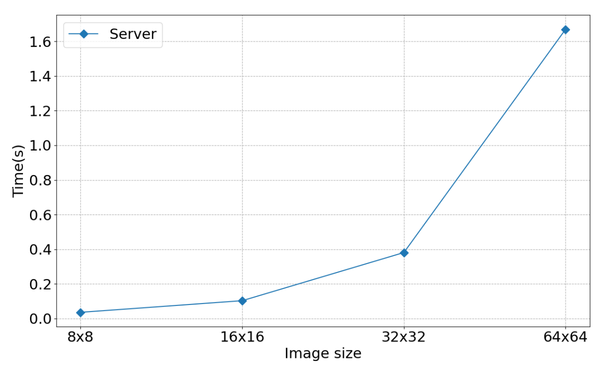

Fig. 9: Classifying time for KNN in a server environment.

B. Edge environment

In this subsection, we describe the results for the edge environment (Raspberry Pi). In this environment, only the classifications for the images were performed, since the edge environment is not suitable for training the models due to the lack of computational resources. Figure 10 shows the classification time of a single image for the CNN model. Similar to the server environment, the classification time increases as the image size increases. Note that the ratio between the longest and the shortest time in this case is only 1.089[s], which is smaller than in the server environment. However, the edge environment was unable to classify 128x128 images due to the lack of RAM.

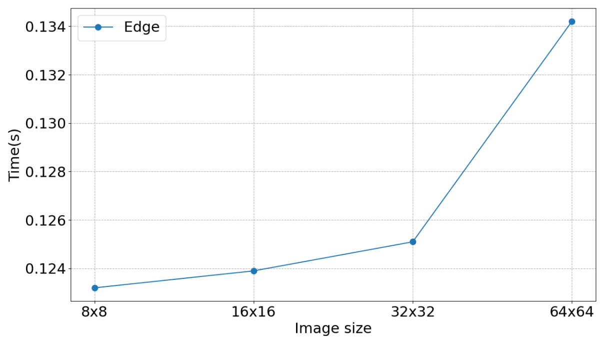

Fig. 10: Time to classify a single image for CNN in an edge environment.

Figure 11 shows the results of the classification for the RF algorithm. Note that it shows only the results for $n_{trees} = 100$. The result for $n\_trees = 500$ is not shown because the edge environment was unable to run the RF model due to lack of memory (RAM). The result shows that the classification time depends on the size of the image. Nevertheless, the ratio between the longest and the shortest time is only 1.03 [s].

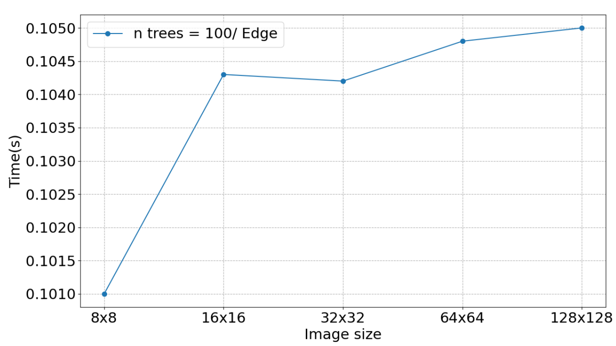

Fig. 11: Time to classify a single image for RF in an edge environment.

Finally, Figure 12 shows the results of the classification time for the KNN model. As expected, the results show that the larger the image size, the longer the classification time. However, the edge did not have enough memory (RAM) to run the KNN model for the 64x64 image size. The ratio between the longest and the shortest time is 8(s), which is quite significant.

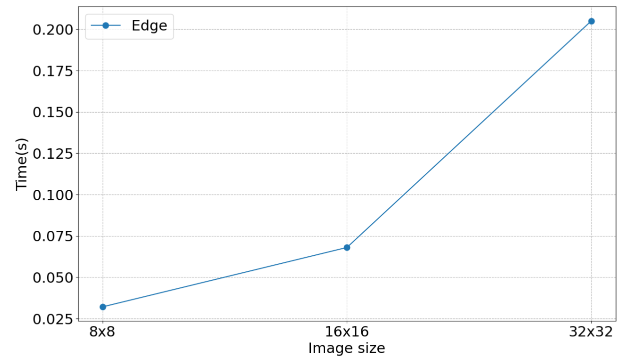

Fig. 12: Single image classifying time for the KNN algorithm executed by the edge Raspberry.

C. Performance Comparison

In this subsection, we present a comparative summary of the results reported in the previous subsections. Table IV summarizes all classification times for each algorithm used in this paper. The first fact to highlight is that there were cases where the edge was not able to classify the images and others that the server was not able to train the model. For these cases, the symbol $-$ was designated, showing that the edge environment was not able to classify or the server environment was not able to train the model. Specifically, the server was unable to train the RF model for 128x128 size images with $n\_trees = 500$ nor the KNN model for 128x128 size images. The edge, on the other hand, was unable to perform a classification for CNN with 128x128 size images, RF with $n\_trees = 500$ and KNN with images of 128x128 and 64x64.

It is also worth to notice that the classification times for RF model, considering the same number of trees, is extremely small. In that way, it is possible to use small size images without worrying about performance. That is, drone-based systems could benefit from this algorithm, since they are resource-constrained. Additionally, for the KNN model, the relationship between Recall and number of trees is not well defined. One possible explanation is that KNN takes into account only a pixel-by-pixel comparison with other images in the data set, being unable to extract features more complex than the pixel value itself.

VI. Conclusions

Drone-based real-time image analysis using ML algorithms is a promising solution to deforestation monitoring for the vast Brazilian Amazon rainforest. Our experimental study revealed that ML-based image classifiers are viable for this purpose even when using low quality images and a poor edge computer which may be equipped with a drone. We analyzed the performance-accuracy trade-offs in three common ML-based classifiers, two different computing environments, and five different quality of images. To investigate the actual performance of drone-based image classification, in our future work, we may also need to consider some environmental

TABLE IV: Classification times

Alg.	size	Server	Edge	N. trees
CNN	8x8	0.07	0.1232	-
	16x16	0.081	0.1239	-
	32x32	0.088	0.1251	-
	64x64	0.113	0.1342	-
	128x128	0.158	-	-
RF	8x8	0.097	0.101	100
	8x8	0.44	-	500
	16x16	0.083	0.1043	100
	16x16	0.486	-	500
	32x32	0.094	0.1042	100
	32x32	0.5	-	500
	64x64	0.086	0.1048	100
	64x64	0.446	-	500
	128x128	0.098	0.405	100
	128x128	-	-	500
KNN	8x8	0.0358	0.0321	-
	16x16	0.103	0.068	-
	32x32	0.381	0.205	-
	64x64	1.67	-	-
	128x128	-	-	-

factors such as the status of wireless communication link, battery usage, and interference from other processes.

ACKNOWLEDGMENT

This research was partially funded by CNPq - Brazil, grant 406263/2018-3.

REFERENCES

[1] A. Voiland, “Tracking amazon deforestation from above,” 2019. [Online]. Available: https://earthobservatory.nasa.gov/images/145988/tracking-amazon-deforestation-from-above

[2] B. São José dos Campos, “Automating land conver change detection: a deep learning based approach to map deforested areas,” Ph.D. dissertation, Instituto Nacional de Pesquisas Espaciais, 2020.

[3] “Global forest change 2000–2018,” 2019. [Online]. Available: https://earthenginepartners.appspot.com/science-2013-global-forest/download_v1.6.html

[4] INPE, “Monitoramento do desmatamento da floresta amazônica brasileira por satélite,” 2020. [Online]. Available: http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/prodes

[5] J. Paneque-Gálvez, M. K. McCall, B. M. Napoletano, S. A. Wich, and L. P. Koh, “Small drones for community-based forest monitoring: An assessment of their feasibility and potential in tropical areas,” *Forests*, vol. 5, no. 6, pp. 1481–1507, 2014.

[6] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

[7] H. Mayfield, C. Smith, M. Gallagher, and M. Hockings, “Use of freely available datasets and machine learning methods in predicting deforestation,” *Environmental modelling & software*, vol. 87, pp. 17–28, 2017.

[8] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[9] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[10] H. Hildmann and E. Kovacs, “Using unmanned aerial vehicles (uavs) as mobile sensing platforms (msps) for disaster response, civil security and public safety,” *Drones*, vol. 3, no. 3, p. 59, 2019.

[11] B. Specktor, “Amazon deforestation shot up by 278% last month, satellite data show,” 2019. [Online]. Available: https://www.livescience.com/66120-amazon-rainforest-deforestation-bolsonaro.html

[12] M. Kanninen, D. Murdiyarso, F. Seymour, A. Angelsen, S. Wunder, and L. German, *Do trees grow on money? The implications of deforestation research for policies to promote REDD*. Cifor, 2007, vol. 4.

[13] S. Dhingra and D. Kumar, “A review of remotely sensed satellite image classification.” *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 9, no. 3, 2019.

[14] M. Pal, “Random forest classifier for remote sensing classification,” *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[15] Planet, “Planet: Understanding the amazon from space,” 2021. [Online]. Available: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data

[16] N. S. Chauhan, “Model evaluation metrics in machine learning,” 2020. [Online]. Available: https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html