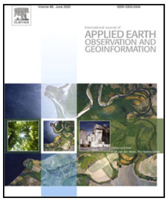
Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag



A deep learning-based mobile application for tree species mapping in RGB images

Mário de Araújo Carvalho[a], José Marcato Junior[b], José Augusto Correa Martins[b], Pedro Zamboni[b], Celso Soares Costa[c], Henrique Lopes Siqueira[b], Márcio Santos Araújo[b], Diogo Nunes Gonçalves[a], Danielle Elis Garcia Furuya[e], Lucas Prado Osco[e], Ana Paula Marques Ramos[e,*], Jonathan Li[d], Amaury Antônio de Castro Junior[a], Wesley Nunes Gonçalves[a,b]

[a] *Faculty of Computer Science, Federal University of Mato Grosso do Sul, Av. Costa e Silva, Campo Grande, 79070-900, MS, Brazil*

[b] *Faculty of Engineering, Architecture, and Urbanism and Geography, Federal University of Mato Grosso do Sul, 79070-900, MS, Brazil*

[c] *Federal Institute of Education, Science and Technology of Mato Grosso do Sul, Ponta Porã 79909-000, MS, Brazil*

[d] *Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

[e] *Program of Environment and Regional Development, University of Western São Paulo, 19067-175, SP, Brazil*

ARTICLE INFO

Keywords:
Convolutional neural networks
Mobile devices
Trees mapping
Remote sensing

ABSTRACT

Tree species mapping is an important type of information demanded in different study fields. However, this task can be expensive and time-consuming, making it difficult to monitor extensive areas. Hence, automatic methods are required to optimize tree species mapping. Here, we propose a deep learning-based mobile application tool for tree species classification in high-spatial-resolution RGB images. Several deep learning architectures were evaluated, including mobile networks and traditional models. A total of 2,349 images were used, of which 1,174 images consisted of the *Dipteryx alata* species and 1,175 images of other local species. These images were manually annotated and randomly divided into training (70%), validation (20%), and testing (10%) subsets, considering the five-fold cross-validation. We evaluated the accuracy and speed (GPU and CPU) of all the implemented deep learning architectures. We found out that the traditional networks have the best performance in terms of F1 score; however, mobile networks are faster. Inception V3 model achieved the best accuracy (F1 score of 97.4%), and MobileNet the worst (F1 score of 83.84%). The MobileNet obtained the best classification speed for CPU (with a mean execution time of 102.8 ms) and GPU (72.4 ms) units. For comparison, Inception V3 achieved a mean execution time of 1058.3 ms for CPU and 634.5 ms for GPU. We conclude that the mobile application proposed can be successfully used to run mobile networks and traditional networks for image classification, but the balance between accuracy and execution time needs to be carefully assessed. This mobile app is a tool for researchers, policymakers, non-governmental organizations, and the general public who intends to assess the tree species, providing a GUI-based platform for non-programmers to access the capabilities of deep learning models in complex classification tasks.

1. Introduction

The expanding market of online mapping, satellite imagery usage, and personal navigation apps have motivated the geospatial community to develop tools to address social and economic demands in multiple contexts. The task of object detection in remote sensing images using machine learning approaches increased and, in the last years, became more accessible to researchers in multiple countries, making it possible to map complex targets using mobile platforms (Ran et al., 2018; Deng, 2019. As a result, mapping tasks have also extended to mobile platforms capable of performing an integrated and dynamic investigation (Giusti et al., 2015) that can support decision-making processes (Giusti et al., 2015; Ran et al., 2018).

Forest monitoring offers essential data to guide public policies on vegetation protection and management, climate change mitigation, and

[*] Corresponding author.
E-mail address: anaramos@unoeste.br (A.P.M. Ramos).

https://doi.org/10.1016/j.jag.2022.103045
Received 7 July 2022; Received in revised form 17 September 2022; Accepted 30 September 2022

1569-8432/Crown Copyright © 2022 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

sustainable development. The capability of detecting individuals or groups of trees is essential for many applications, such as resource inventories, wildlife habitat mapping, biodiversity assessment, and threat and stress management (Fassnacht et al., 2016). For urban centers, where the deteriorating ecological environment is critical in urban ecosystems (Bastian et al., 2012), it is crucial to map vegetation to assist practices that aim to improve its sustainability and resilience. Trees provide an array of ecosystem services, including regulating and maintaining local climatic conditions by reducing the formation of heat islands and increasing water infiltration in the soil. They also nourish habitat for local biodiversity, provisioning resources (wood, food, and biomass), cultural, economic, and historical values, and scenic landscapes (Baró et al., 2014; McHugh et al., 2015). Therefore, the preservation of these systems requires accurate and precise data gathering, especially for the more sensitive or endangered tree species. However, individual tree mapping based on ground surveys can be expensive and time-consuming, evolving specialists in this field, making the task difficult to be implemented for larger areas.

Remote sensing platforms are essential for data acquisition at different scales, like orbital, suborbital, aerial, and terrestrial levels (Alonzo et al., 2014). Advances in Unmanned Aerial Vehicles (UAV) platforms technologies (Salamí et al., 2014), for example, enable acquiring aerial images at a high spatial resolution rapidly and at a lower cost than in the past. High spatial-resolution imagery use enables a more refined data analysis required in some applications like those related to tree species mapping (Lobo Torres et al., 2020; Liu et al., 2017). Nevertheless, due to the large amount of data generated, there is a demand for building tools that can efficiently process this information. As such, integrating deep learning methods and mobile platforms may result in a robust and low-cost approach to deal with this issue. Among the deep learning architectures, the usage of convolutional neural networks (CNN) based architectures is currently the most used approach to extract information from remote sensing imagery (Martins et al., 2021; dos Santos et al., 2019; Osco et al., 2019; Li et al., 2019).

For tree species mapping issues, several studies have combined deep learning methods and remote sensing data such as multi and hyperspectral images, LiDAR, and a combination of both (Ferreira et al., 2020; Torabzadeh et al., 2019; Franklin and Ahmed, 2018; Xie et al., 2019). However, classifying tree species in high spatial-resolution RGB imagery with deep learning models should be encouraged since RGB sensors are cheaper and frequently embedded in UAV platforms. Additionally, in recent years, there has been a growing availability of powerful and low-cost mobile phones and devices that researchers have explored with deep learning methods (Sandler et al., 2018; Ran et al., 2018; Deng, 2019; Suharjito et al., 2021).

Mobile applications for tree mapping using CNNs are still poorly explored. This is due to the fact that algorithms of this type of approach are computationally expensive, especially for mobile devices that do not have robust hardware when in comparison to desktop computers. For video-logging systems, mobile mapping phase systems can offer full 3-D mapping capabilities that are achieved using advanced multi-sensor integrated data acquisition and processing technology, making it possible to refine data analysis (Nyqvist et al., 2020; Zhang et al., 2020; Sant'Ana et al., 2021; Qian et al., 2021). As highlighted by Tao and Li (2007), the recent technological trend in mobile mapping can be characterized by: (1) the increasing usage of mobile and portable sensors with low-cost, enabling direct georeferencing appliances; and (2) a collaborative mapping with networked multi-platform sensors. Thus, even though mobile phones have considerably less hardware power than desktop computers, they could be a suitable platform to run CNN's models.

In the aforementioned context, this study proposes a deep learning-based mobile application for tree species classification in RGB imagery with high spatial resolution. We considered CNNs designed specifically for mobile devices, such as the MobileNet V2 (Sandler et al., 2018) and NASNet Mobile (Zoph et al., 2017), and traditional CNNs like InceptionResNet V2 (Szegedy et al., 2016), Inception V3 (Szegedy et al., 2015), and ResNets (He et al., 2016). For our study case, we trained these CNNs to map the *Dipteryx alata* tree species, popularly known as Cumbaru or Baru in their native regions. The *Dipteryx alata* trees are common in the Cerrado biome regions in Brazil and have important ecological and socioeconomic value (Martins, 2010). Moreover, we analyzed the computational cost of these CNNs when executed on mobile devices, calculating the mean execution time in both the CPU and the GPU. Our research fills the literature gap related to the usage of deep learning on mobile phones as a platform for ecological remote sensing applications and provides an easy-to-use application tool for mapping tree species.

Table 1
Mobile device specification.

Smartphone	Moto Z2 Play
Screen resolution	1080 × 1920
Camera	12 Mp/4032 × 3024
ChipSet	Snapdragon 626
RAM	4 GB
Android version	9.0

2. Materials and methods

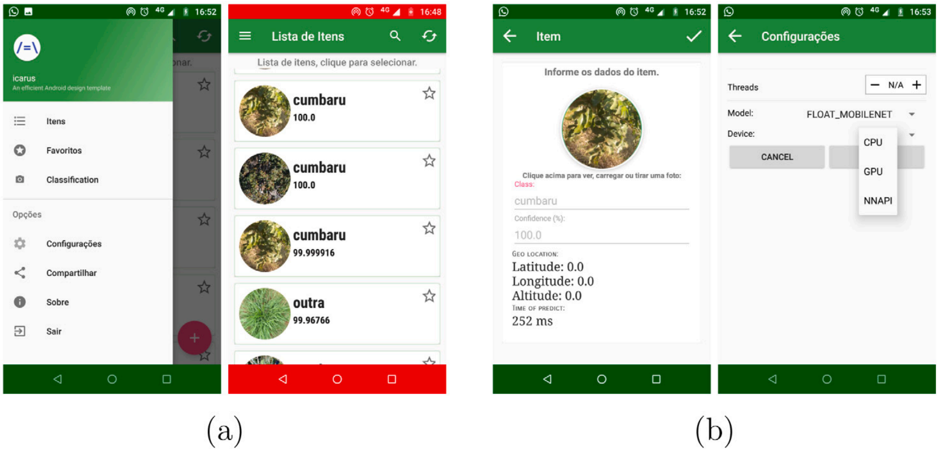
2.1. Generic mobile application for remote sensing

For this study, a tool for classifying remote sensing imagery was developed, and the tests proved that it enables more agility to map tree species. The developed application has a modern design, extra features for image pre-processing, a database to store the ratings and captured images, real-time rating using the device's camera, rating new photos, or uploading an image from the gallery, and the storage of the georeferenced image with the collection of latitude, longitude, and altitude. We developed the application to accept all classification models tested on Tensorflow with useful source code programming practices to facilitate maintenance, extension, and calculation of inference processing time. The application was named iCarus by the development team, and its complete source code and other artifacts are freely available in the Geomatica laboratory source code repository at the following link: https://gitlab.com/geomatics-laboratory/deep-learning/classification/icarus. Fig. 1 shows screenshots of the iCarus application interface.

The tests were performed on a smartphone equipped with a 12 Mp/4032 × 3024 (Table 1) where the iCarus application was installed and tested. For a better assessment, experiments on the mobile device were performed on both the CPU and GPU. A configuration option was implemented in iCarus, where it is possible to change the application to make inferences on the CPU or GPU. The configuration of the number of CPU threads in the application is also available. All CPU tests were performed with 4 threads.

2.2. Project development workflow

Our workflow was divided into seven main stages (Fig. 2). (a) Acquisition of RGB images by a UAV flight carried out in the metropolitan region of Campo Grande, State of Mato Grosso do Sul, Brazil, followed by the annotation of trees in the images; (b) Division of images into training, validation, and test subsets; (c) Definition of five state-of-the-art deep learning architectures for the classification of *Dipteryx alata* trees; (d) Architecture training, application of transfer learning and hyperparameter adjustment with cross-validation; (e) Architecture evaluation with metrics such as F1 score, confusion matrix, inference time, and application of ANOVA and Tukey tests; and (f) Deployment of architectures through an Android application and performance evaluation in terms of accuracy and execution time in CPU and GPU.



Class: cumbaru
Confidence (%): 100.0
Geo location:
Latitude: 0.0
Longitude: 0.0
Altitude: 0.0
Time of predict:
252 ms

Threads N/A
Model: FLOAT_MOBILENET
Device:

Fig. 1. Screenshots of the iCarus application working: (a) Menu options and screen with the cards of the inferences made; (b) Screen for capturing the images and setting the parameters of the application.

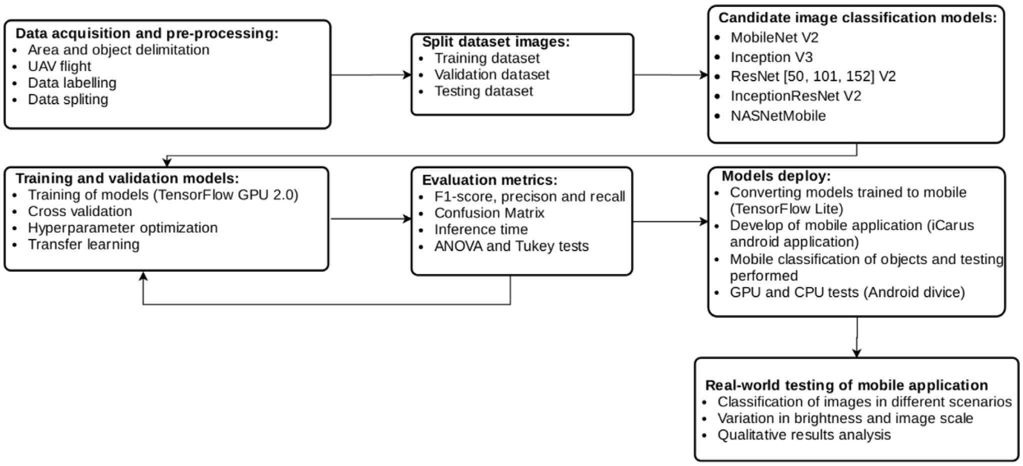

Data acquisition and pre-processing:
Area and object delimitation
UAV flight
Data labelling
Data spliting
Split dataset images:
Training dataset
Validation dataset
Testing dataset
Candidate image classification models:
MobileNet V2
Inception V3
ResNet [50, 101, 152] V2
InceptionResNet V2
NASNetMobile
Training and validation models:
Training of models (TensorFlow GPU 2.0)
Cross validation
Hyperparameter optimization
Transfer learning
Evaluation metrics:
F1-score, precison and recall
Confusion Matrix
Inference time
ANOVA and Tukey tests
Models deploy:
Converting models trained to mobile (TensorFlow Lite)
Develop of mobile application (iCarus android application)
Mobile classification of objects and testing performed
GPU and CPU tests (Android divice)
Real-world testing of mobile application
Classification of images in different scenarios
Variation in brightness and image scale
Qualitative results analysis

Fig. 2. Workflow summarizing the method implemented in our study.

Table 2
UAV equipment and flight specifications.

Aircraft	Phantom 4 Advanced
Sensor	1″ CMOS
Field of view	85°
Nominal focal length	8.8 mm
Image size	5472 × 3648
Mean GSD	0.85 cm
Mean flight height	35 m

2.3. Study area and data acquisition

The study was conducted in the urban area of the city of Campo Grande, inside the state of Mato Grosso do Sul, which is located in the Cerrado Biome of Brazil as presented in Fig. 3. A total of 2349 images were acquired in between five months (January–March 2019 and September–December 2019), with five missions. An advanced Phantom 4 UAV equipped with a 20-megapixel RGB camera captured images at 25–45 m flight heights (Table 2). Images have a resolution of 5472 pixels by 3648 pixels and a Ground Sample Distance (GSD) of 0.85 cm (centimeter) and were captured inside two regions in the urban part of the municipality, totaling an area of approximately 95 ha. Approximately 75 Cumbaru trees were photographed during these missions.

Examples of tree image samples that were captured via UAV are shown in Fig. 4. It is important to note that the images of the trees were captured at different times of the year and with various conditions, such as climate, appearance, scale, lighting, leaf color, presence of fruits, and presence of other trees in their surroundings. This data variability provided a challenging scenario for our research, which is important for monitoring studies in real-life situations. All captured images were manually labeled by specialists using the LabelMe open annotation tool software (http://labelme.csail.mit.edu/). In this process, *Dipteryx alata* trees were annotated with bounding boxes. This dataset was properly annotated and can be used both in experiments of image classification and in object detection. As the objective of this study was to evaluate classification methods, the images were cropped to the bounding boxes and resized to 1024 × 1024 pixels. Samples of the cropped images containing the target species and other tree species are shown in Fig. 4. The classification dataset has 1174 images of *Dipteryx alata* and 1175 images of other local species in the Cerrado Biome. The description of how this subset of images was partitioned for training and testing is described in detail in Section 2.5, including details of cross-validation and data augmentation.

2.4. Image classification models

A total of five deep learning architectures were evaluated in this study for benchmark purposes of tree species classification using mobile platforms. We categorize them into two groups: traditional networks - TN (Inception V3, ResNet, and InceptionResNet V2) and mobile networks - MN (MobileNet and NASNets). It is important to note that

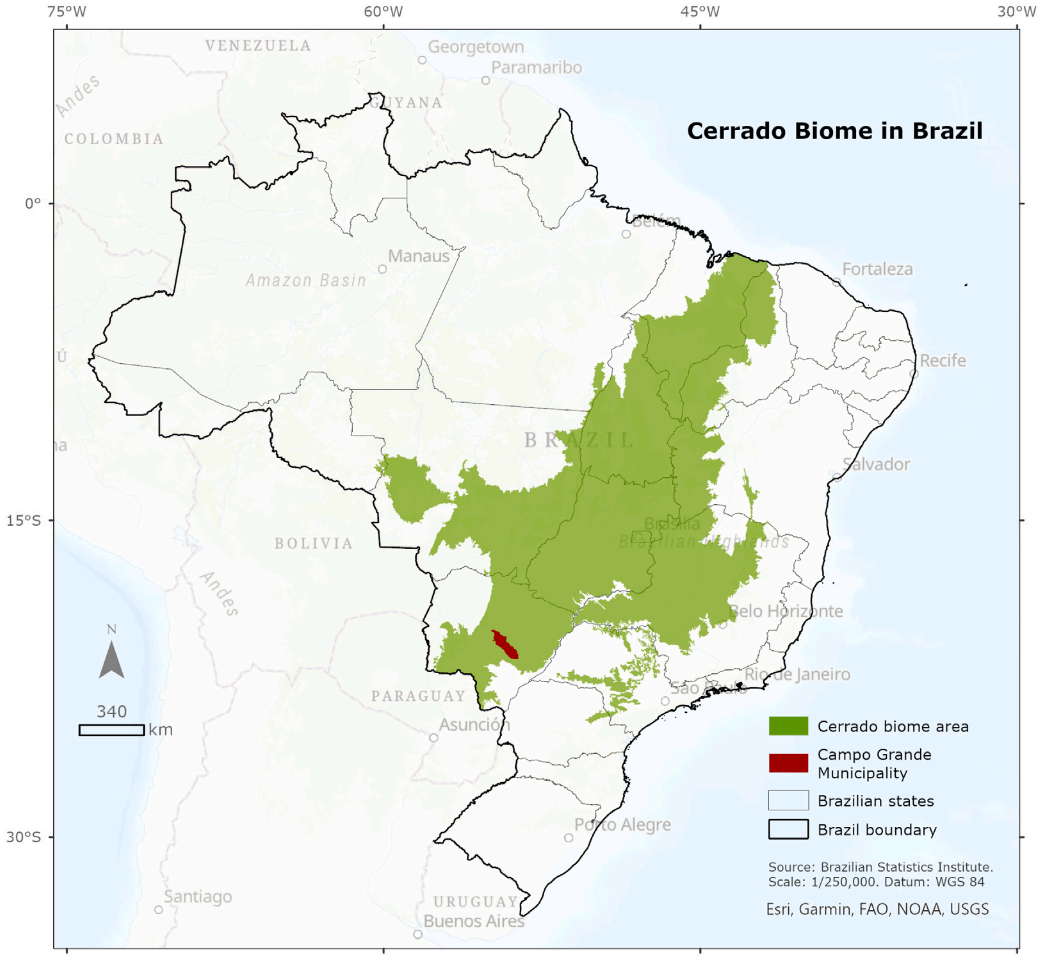


Fig. 3. Spatial distribution of the Cerrado biome in Brazil and Campo Grande municipality where our study area is located.

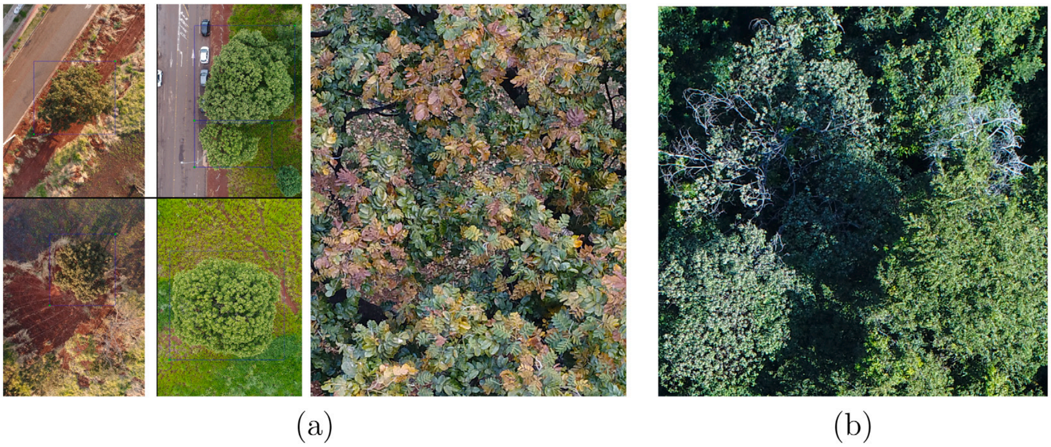


Fig. 4. Examples of images cropped from the image subset; (a) *Dipteryx alata* tree images on different dates; (b) Other tree species from the dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

traditional networks indicate that they are methods designed to process data in desktop machines, with powerful Graphical Processing Units (GPU), and mobile networks are designed to run in smaller and lesser computationally demanding systems. The image classification models used in this study are briefly described below.

- Inception V3 (Szegedy et al., 2015) evaluates how an ideal local sparse structure of a network can be approximated and covered by dense components. The motivation for this is to increase the size of the network with a reasonable processing cost, as it is known that the most direct way to improve accuracy is to increase its size. In Inception V3, the authors proposed a new approach to the initial architecture. For this, they explore other ways of factorizing convolutions in various settings and use regularization techniques.
- ResNet (He et al., 2016) is a CNN winner of the 2015 ILSVRC. This architecture uses residual learning to train deeper networks.

This approach is motivated by the phenomenon known as degradation when the learning in the deeper layers becomes saturated. Therefore, the authors proposed to add the input of a block with the output according to Eq. (1), where V_e and $F(V_e)$ are the input and output, and F is a set of two convolution layers.

$$V_s = F(V_e) + V_e \tag{1}$$

- InceptionResNet V2 (Szegedy et al., 2016) presents a combination of residual learning and inception architecture (inception module). The idea is to have residual learning after an inception module. Training with residual connections accelerates the training of inception networks significantly, as shown in Szegedy et al. (2016), which provides evidence that combining networks can increase the performance and accuracy of image classification problems.
- MobileNet (Sandler et al., 2018) improves the performance of mobile models in several aspects, providing greater performance in the prediction time and size of the trained models. This model works with residual blocks, where both the input and the output are thin neck layers opposite to traditional residual models. This made it possible for light deep convolutions to filter features in the intermediate expansion layers. To maintain the representational power in the model, non-linearity in the narrow layers was removed to improve performance. The performance of the model was tested in the ImageNet (Deng et al., 2009) dataset for classification, COCO for object detection, and VOC for image segmentation.
- NASNets (Zoph et al., 2017) proposes a learning architecture directly from its dataset. Although this technique is costly, its accuracy places NASNet Large among the state-of-the-art classification CNNs. NASNet proposes to search for an architectural building block in a smaller dataset and then transfer this block to the larger one. This process allows NASNet to work with content transferability. NASNet Mobile, a reduced version, is 3.1% better than modern mobile CNNs, compared to the equivalent size for mobile platforms. Due to the cost of the hardware to train the NASNet Large, the experiments in this study were performed only with NASNet Mobile.

2.5. Experimental setup

In our experiments, the five-fold cross-validation strategy was adopted to obtain a more reliable method of validating the models. This cross-validation strategy was applied to ensure that all images are used in the test. The weights of all CNNs were initialized with pre-trained values from the ImageNet dataset (http://www.image-net.org/). This strategy is known as transfer learning, whose objective is to take advantage of the weights trained in a large database to accelerate the feature extraction process in the new application. With cross-validation evaluation, the average accuracy of the five-folds can be obtained while the standard deviation can indicate the possible discrepancies in the obtained accuracy values. Adam optimizer was used in the training phase of the models. Throughout several empirical tests using learning rates of 0.01, 0.001, and 0.0001, it was demonstrated that the convergence of the loss function obtained better results when using a learning rate of 0.001. Fig. 5 shows the accuracy and loss curves for the MobileNet V2.

As noted, the loss function declines rapidly in the first epochs and stabilizes, indicating that the number of epochs has been sufficient and the learning rate is adequate. Furthermore, the accuracy in the validation set has not increased in the last five epochs. Therefore, the learning rate was set to 0.001 and the number of epochs to 30 for all models.

To reduce overfitting during training, the data augmentation technique, including re-scale, rotation, flip, shift, shear, and zoom, was applied to the images during the training process. The combination of data augmentation and cross-validation techniques allows to train and evaluate models more reliably during training, the main idea of this combination is to use the generation of unseen images for the training set so that the same does not memorize the dataset and reduce the risk of overfitting. The training and validation of the methods were carried out on a workstation computer equipped with an Intel®Xeon CPU E3-1270 @ 3.80 GHz, 250 GB SSD with 64 GB of RAM, Titan V graphics card with 12 GB of graphics memory, CUDA version 10.2, and Ubuntu 20.04 operating system. We also exported and validated the models using the Tensorflow Lite library, which enabled the development of the application called iCarus for testing the models on mobile devices.

2.6. Performance metrics

The performance of the models was evaluated using the metric F1 score (F1). The F1 score metric is calculated based on the weighted average of Precision (P) and Recall (R), where an F1 score reaches its best value at 1 and the worst score at 0. The precision (P) metric is defined as the number of True Positives (TP) divided by the number of True Positives (TP) plus the number of False Positives (FP), as shown in Eq. (2). The recall (R) metric is defined as the number of True Positives (TP) over the number of True Positives (TP) plus the number of False Negatives (FN), as shown in the (3) equation. The equation for the F1 score (F1) is described in Eq. (4).

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

For performance analysis on the mobile device, we calculated the average processing time of each model, in both the CPU and GPU, for image prediction. Each model was exported to the iCarus application mobile, where tests were carried out to verify prediction in a set of test images. In this sense, the loading times of the models and data preprocessing were discarded. Only the processing time for the prediction of each image was considered in the analyses.

A one-way ANOVA test was applied to check if there were differences between the methods analyzed at a significance level of 5%, using the F1 score metric as the independent variable. After that, Tukey's post hoc test was applied to the results of the classification models to identify the statistical differences between the deep learning methods tested. The results were analyzed using descriptive statistics, with a boxplot graph, to verify the models' convergence of losses and accuracy.

3. Results and discussion

This section presents the results obtained by the models and their discussion. In Section 3.1 the results on the desktop computer are presented, and in Section 3.2 is presented the performance on the mobile devices.

3.1. Experiments on desktop computer

The F1 scores of the classification of *Dipteryx alata* tree species using deep learning models in each cross-validation round (R1–R5) are presented in Table 3. The last column of this table presents the average of the rounds, with Inception V3 having the highest value. ResNet101 with 94.6% and InceptionResNet with 93.8% have the second and third best F1 score averages, respectively. The results of the ANOVA test indicated a p-value of 6.3647e^{-7}, being less than the significance level ($\alpha = 0.05$). Therefore, we can reject the null hypothesis that the F1 score averages are similar. According to the post hoc Tukey's test, CNNs can be categorized into three accuracy categories (a), (b), and

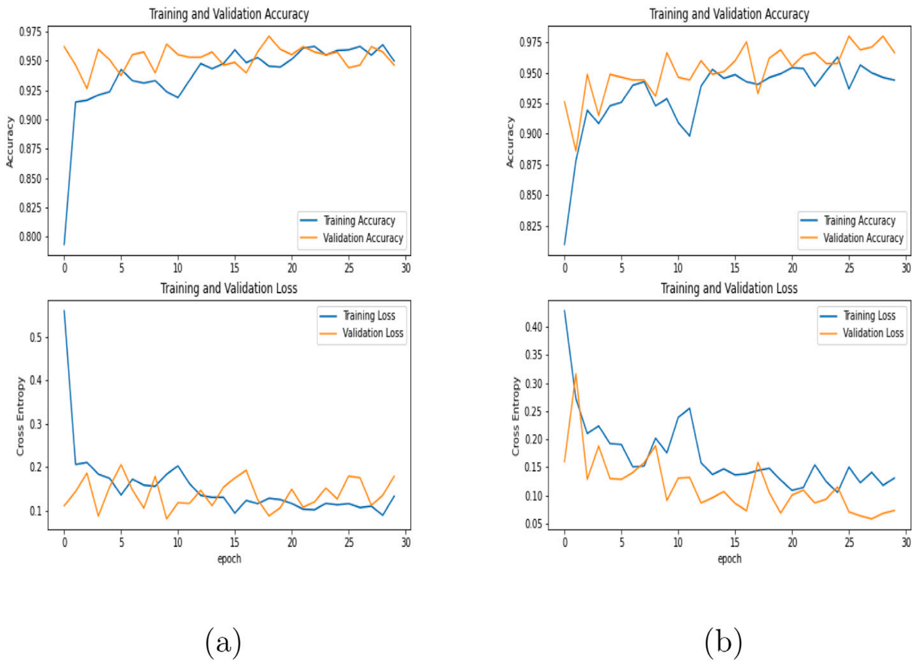


Fig. 5. Accuracy and loss curves for training and validation phases for the MobileNet V2 network.

Table 3
Average F score metric for the *Dipteryx alata* tree classification in five cross-validation rounds (R1–R5).

Model	R1	R2	R3	R4	R5	F score (std)
InceptionResNet V2	91.00	91.00	97.00	97.00	93.00	93.8
Inception V3	98.00	97.00	97.00	98.00	97.00	**97.4**
MobileNet V2	87.00	76.00	79.00	89.00	88.00	83.8
NASNet Mobile	88.00	90.00	90.00	92.00	90.00	90.0
ResNet101 V2	94.00	97.00	94.00	95.00	93.00	94.6
ResNet152 V2	93.00	94.00	93.00	93.00	95.00	93.6
ResNet50 V2	93.00	90.00	91.00	90.00	93.00	91.4

Table 4
Mean execution time of all the classification models using the CPU and GPU of the mobile device.

Model	Mean Exec. time (s)	Stand. Dev. (ms)	Device
InceptionResNet V2	2.2511 1.4245	20.3492 6.7564	CPU GPU
Inception V3	1.0583 0.6345	15,8811 4.1048	CPU GPU
MobileNet V2	**0.1028** **0.0724**	3.0919 1.9078	CPU GPU
NASNet Mobile	0.2561 0.3972	1.2206 5.3814	CPU GPU
ResNet101 V2	1.3897 0.6697	39.6839 2.4104	CPU GPU
ResNet152 V2	2.1323 0.9719	2.2925 3.4771	CPU GPU
ResNet50 V2	0.7172 0.3622	24.9471 8.1216	CPU GPU

(c). These categories are represented by Inception V3 (a), ResNet50 and NASNet Mobile (b), and MobileNet V2 (c). Although categories (a) and (b) present significant differences between them, they did not present significant differences in relation to the CNNs of the category (ab), composed by the InceptionResNet, ResNet101, and ResNet152. MobileNet V2 is a CNN optimized for mobile devices, and therefore its F1 score was expected to be smaller than all the heavier and more complex CNNs. However, it is important to point the F1 score obtained by NASNet Mobile, which was assigned to group (b) with 90.0.

Fig. 6 shows boxplots with the performance achieved by each CNN. When analyzing the distribution of the metric, we noticed that Inception V3 has the most negligible dispersion around the median. The others have lower performance compared to Inception V3, in addition to more dispersion around the median and the presence of outliers. MobileNet presents the highest dispersion along with the worst performance.

3.2. Experiments on mobile device

This subsection presents the run-time experiments on mobile devices. Table 4 shows the average time in seconds using the CPU and GPU of the device. The average time was obtained by performing the classification in 30 randomly chosen images.

MobileNet obtained the lowest computational cost with similar values for GPU (0.0724 s) and CPU (0.1028 s) when compared to the differences in computational costs among GPU and CPU observed for the other evaluated networks (Table 4). The second and third lowest computational costs were NASNet Mobile and ResNet50, respectively. These results were expected since MobileNet and NASNet Mobile are designed and optimized to work on mobile devices, which implies less complexity in their architecture. The computational cost of MobileNet V2 was up to 10 times faster than the InceptionV3, thus demonstrating its effectiveness and efficiency in mobile devices. The CPU and GPU times also had differences. In general, the execution through GPU obtains an increase in processing speed. This can be observed in the experiments, except with NASNet Mobile, which was the only CNN that obtained a lower result on the GPU. The results showed that specialized mobile models are simple, and the GPU usage does not make a significant difference as it does for the general models.

To test the generalization capability and robustness of the models, we performed the prediction using images under different conditions. As aforementioned, we captured images at different times of the year to carry out tests with different scenarios (variation of lighting, scale, presence of fruits, presence of flowers, and leaves with different shades of colors). Despite the different variations of the images, all the models tested achieved good performance in their classification as presented in Fig. 7.

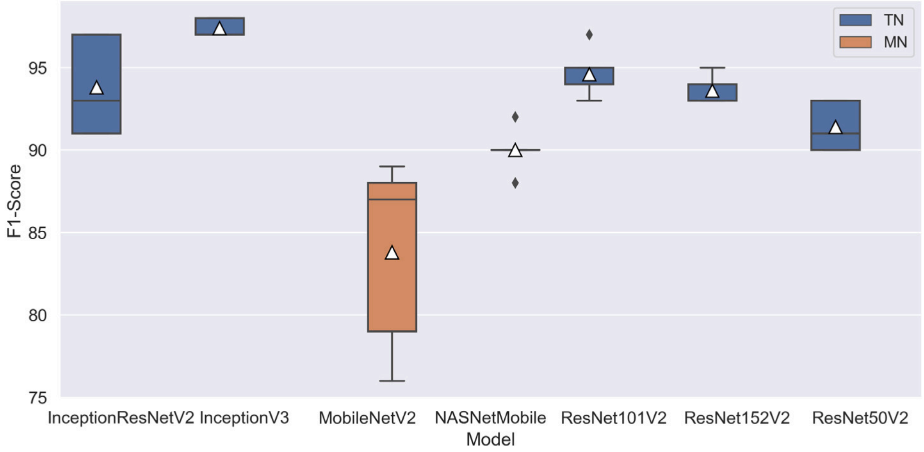
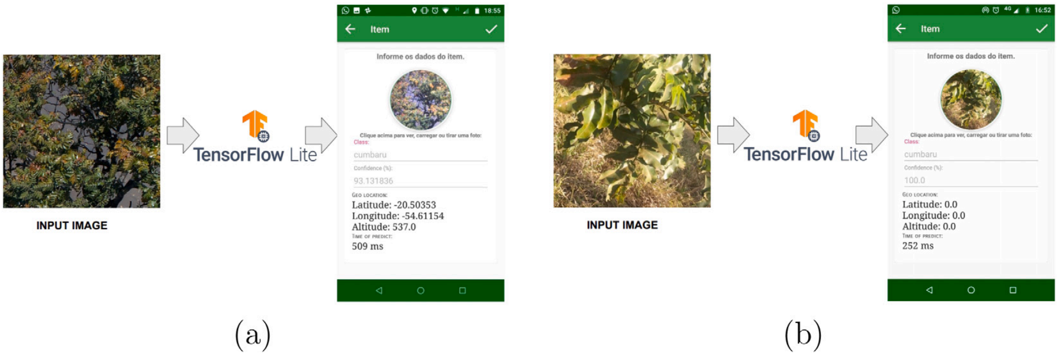


Fig. 6. Boxplot comparing the performance of the models using F1 score. TN and MN stand for Traditional Networks and Mobile Networks, respectively.



Informe os dados do item.
Clique acima para ver, carregar ou tirar uma foto:
Class:
cumbaru
Confidence (%):
93.131836
Geo location:
Latitude: -20.50353
Longitude: -54.61154
Altitude: 537.0
Time of predict:
509 ms

Informe os dados do item.
Clique acima para ver, carregar ou tirar uma foto:
Class:
cumbaru
Confidence (%):
100.0
Geo location:
Latitude: 0.0
Longitude: 0.0
Altitude: 0.0
Time of predict:
252 ms

Fig. 7. (a) NASNet Mobile classification on GPU; (b) Predictions in the iCarus application. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We noticed a trade-off in terms of speed and F score in our tests. The choice of the best model is complex, and it needs to consider the importance of the F score and the time cost of the application. The overall best F score and time cost models are Inception V3 and MobileNet V2, respectively. MobileNet V2, the model with the best results in terms of time cost, showed the worst F score. On the other hand, Inception V3, the model with the best results in F score, did not result in satisfactory results in comparison with the time cost. Comparing MobileNet with Inception V3, the difference is 955.5 ms for the CPU and 562.1 ms for the GPU. This method had the best results compared to InceptionResNet V2, ResNet101 V2, and ResNet152 V2. The Inception V3 was surpassed, in terms of execution time, only by networks specialized in mobile.

When choosing the best network to perform the current classification task, the user should choose the best metric that is available on the app. Despite the difference in time, the fastest CNN MobileNet V2 takes 0.07 s in a GPU and 0.01 s in a CPU, and the slowest CNN InceptionResNet V2 takes 1.42 s in a GPU and 2.25 in a CPU, we are talking of no more than 2 s of difference in time, which may not be a problem in most of the application that needs a fast classification response.

We notice a trade-off in terms of speed and F1 score in our tests (Fig. 8). The choice of the best model is complex, and the user needs to consider the importance of the F1 score and the execution cost for the application. Specialized models for mobile devices showed better performance in terms of speed but lower results in terms of the F1 score. The other models present better F1 scores but also higher execution times.

The overall best F1 scores and execution cost models are Inception V3 and MobileNet, respectively. On the other hand, MobileNet presented the worst F1 while Inception V3 did not get satisfactory execution cost. Comparing MobileNet with Inception V3, the difference is 955.5 ms for the CPU and 562.1 ms for the GPU. However, the Inception V3 model performance was surpassed, in terms of execution time, only by the CNN specialized for mobile devices.

In ecology, classification algorithms are among the most extensively used statistical tools. Ecological data is frequently multidimensional, with nonlinear and complicated interactions between variables, as well as a large number of missing values among measured variables. Having a toll that classifies this data with an easy-to-use implementation in mobile devices is a step forward in terms of environmental assessments, supporting the creation of policies that improve the health of human and vegetation systems.

4. Conclusion

Here we investigated tree monitoring with remote sensing data in combination with deep learning methods and mobile device image capture. One of the most complex aspects of forest management is to determine the precise number of trees and the corresponding species. We were able to perform this task with success by implementing the proposed method. Five deep learning architectures were evaluated, three traditional and two mobile networks. We evaluated the models in terms of F1 score and execution time. We also provided statistical analysis for the evaluation metrics used. Our results indicated that all CNNs show satisfactory performance. Traditional CNNs have better F1 scores and mobile CNNs have a shorter runtime. Our experimental results demonstrated that mobile phones are suitable platforms to run both traditional and mobile deep learning models for image classification

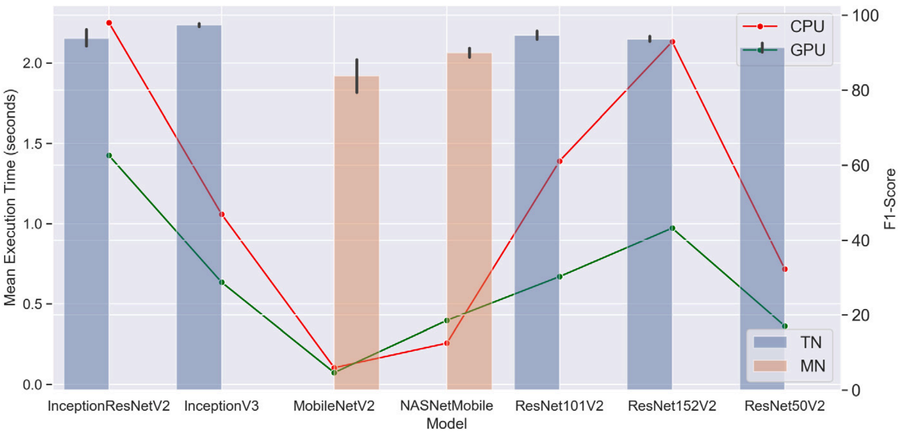


Fig. 8. Bar plot for F1 score and line plot for Mean Execution Time using CPU and GPU for all the models.

purposes. However, for the choice of the model, the user must consider the trade-off between accuracy and speed.

In conclusion, this study demonstrates the potential of combining deep learning methods along with mobile phones for image classification problems. Once mobile phones became cheaper and more accessible to the general public, our methodology can be helpful for research, assisting governmental and non-governmental organizations to map tree species or other classification tasks. For future works, we intend to make the application tool available for public usage, train the network to classify more tree species and improve our models to reach better accuracy and time results, making it a viable approach to the automatic classification of trees species at the same moment that the image is captured by the mobile device and send to its CPU or GPU, providing real-time application.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The authors would like to acknowledge Nvidia Corporation for the donation of the Titan X graphics card.

Funding

This research was funded by the Scientific Initiation Program (PIBIC) of the Federal University of Mato Grosso do Sul (UFMS) . This research was funded by CNPq, Brazil (p: 433783/2018-4, 310517/2020-6, 314902/2018-0, 304052/2019-1 and 303559/2019-5), FUNDECT, Brazil (p: 59/300. 066/2015, 071/2015) and CAPES PrInt, Brazil (p: 88881.311850/2018-01, 88887.580307 2020-00, 88887.635596 2021-00). The authors acknowledge the support of the UFMS (Federal University of Mato Grosso do Sul), Brazil and CAPES, Brazil (Finance Code 001). This research was also partially supported by the Emerging Interdisciplinary Project of Central University of Finance and Economics, China.

References

Alonzo, M., Bookhagen, B., Roberts, D., 2014. Urban tree species mapping using hyperspectral and LiDAR data fusion. Remote Sens. Environ. 148, 70–83.

Baró, F., Chaparro, L., Gómez-Baggethun, E., Langemeyer, J., Nowak, D.J., Terradas, J., 2014. Contribution of ecosystem services to air quality and climate change mitigation policies: The case of urban forests in Barcelona, Spain. Ambio http://dx.doi.org/10.1007/s13280-014-0507-x.

Bastian, O., Haase, D., Grunewald, K., 2012. Ecosystem properties, potentials and services – The EPPS conceptual framework and an urban application example. Ecol. Indic. 21, 7–16.

Deng, Y., 2019. Deep learning on mobile devices: a review. In: Mobile Multimedia/Image Processing, Security, and Applications 2019, Vol. 10993. International Society for Optics and Photonics, 109930A.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Fassnacht, F.E., Latifi, H., Stereńczak, K., Modzelewska, A., Lefsky, M., Waser, L.T., Straub, C., Ghosh, A., 2016. Review of studies on tree species classification from remotely sensed data. Remote Sens. Environ. 186, 64–87. http://dx.doi.org/10.1016/j.rse.2016.08.013.

Ferreira, M.P., de Almeida, D.R.A., de Almeida Papa, D., Minervino, J.B.S., Veras, H.F.P., Formighieri, A., Santos, C.A.N., Ferreira, M.A.D., Figueiredo, E.O., Ferreira, E.J.L., 2020. Individual tree detection and species classification of Amazonian palms using UAV images and deep learning. Forest Ecol. Manag. 475, 118397. http://dx.doi.org/10.1016/j.foreco.2020.118397, URL: https://www.sciencedirect.com/science/article/pii/S037811272031166X.

Franklin, S.E., Ahmed, O.S., 2018. Deciduous tree species classification using object-based analysis and machine learning with unmanned aerial vehicle multispectral data. Int. J. Remote Sens. 39 (15–16), 5236–5245.

Giusti, A., Guzzi, J., Cireşan, D.C., He, F.-L., Rodríguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al., 2015. A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robot. Autom. Lett. 1 (2), 661–667.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., Benediktsson, J.A., 2019. Deep learning for hyperspectral image classification: An overview. IEEE Trans. Geosci. Remote Sens. 57 (9), 6690–6709. http://dx.doi.org/10.1109/TGRS.2019.2907932.

Liu, L., Coops, N.C., Aven, N.W., Pang, Y., 2017. Mapping urban tree species using integrated airborne hyperspectral and LiDAR remote sensing data. Remote Sens. Environ. 200, 170–182. http://dx.doi.org/10.1016/j.rse.2017.08.010, URL: https://www.sciencedirect.com/science/article/pii/S0034425717303620.

Lobo Torres, D., Queiroz Feitosa, R., Nigri Happ, P., Elena Cué La Rosa, L., Marcato Junior, J., Martins, J., Olã Bressan, P., Gonçalves, W.N., Liesenberg, V., 2020. Applying fully convolutional architectures for semantic segmentation of a single tree species in urban environment on high resolution UAV optical imagery. Sensors 20 (2), http://dx.doi.org/10.3390/s20020563, URL: https://www.mdpi.com/1424-8220/20/2/563.

Martins, B., 2010. Desenvolvimento Tecnológico Para o Aprimoramento do Processamento de Polpa e Amêndoa de Baru (Dipteryx Alata Vog) (Ph.D. thesis). Faculty of Food Engineering (FEA) – University of Campinas, http://repositorio.unicamp.br/jspui/handle/REPOSIP/256419.

Martins, J.A.C., Nogueira, K., Osco, L.P., Gomes, F.D.G., Furuya, D.E.G., Gonçalves, W.N., Sant'Ana, D.A., Ramos, A.P.M., Liesenberg, V., dos Santos, J.A., de Oliveira, P.T.S., Junior, J.M., 2021. Semantic segmentation of tree-canopy in urban environment with pixel-wise deep learning. Remote Sens. 13 (16), http://dx.doi.org/10.3390/rs13163054, URL: https://www.mdpi.com/2072-4292/13/16/3054.

McHugh, N., Edmondson, J.L., Gaston, K.J., Leake, J.R., O'Sullivan, O.S., 2015. Modelling short-rotation coppice and tree planting for urban carbon management - a citywide analysis. J. Appl. Ecol. http://dx.doi.org/10.1111/1365-2664.12491.

Nyqvist, D., Hedenberg, F., Calles, O., Österling, M., von Proschwitz, T., Watz, J., 2020. Tracking the movement of PIT-tagged terrestrial slugs (arion vulgaris) in forest and garden habitats using mobile antennas. J. Mollusc. Stud. 86 (1), 79–82.

Osco, L.P., Marques Ramos, A.P., Saito Moriya, É.A., de Souza, M., Marcato Junior, J., Matsubara, E.T., Imai, N.N., Creste, J.E., 2019. Improvement of leaf nitrogen content inference in Valencia-orange trees applying spectral analysis algorithms in UAV mounted-sensor images. Int. J. Appl. Earth Obs. Geoinf. 83, 101907. http://dx.doi.org/10.1016/j.jag.2019.101907, URL: https://www.sciencedirect.com/science/article/pii/S030324341930491X.

Qian, J., Chen, K., Chen, Q., Yang, Y., Zhang, J., Chen, S., 2021. Robust visual-lidar simultaneous localization and mapping system for UAV. IEEE Geosci. Remote Sens. Lett..

Ran, X., Chen, H., Zhu, X., Liu, Z., Chen, J., 2018. Deepdecision: A mobile deep learning framework for edge video analytics. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, pp. 1421–1429.

Salamí, E., Barrado, C., Pastor, E., 2014. UAV flight experiments applied to the remote sensing of vegetated areas. Remote Sens. 6 (11), 11051–11081. http://dx.doi.org/10.3390/rs61111051, URL: https://www.mdpi.com/2072-4292/6/11/11051.

Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L., 2018. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR abs/1801.04381. arXiv:1801.04381. URL: http://arxiv.org/abs/1801.04381.

Sant'Ana, D.A., Pache, M.C.B., Martins, J., Soares, W.P., de Melo, S.L.N., Garcia, V., de Moares Weber, V.A., da Silva Heimbach, N., Mateus, R.G., Pistori, H., 2021. Weighing live sheep using computer vision techniques and regression machine learning. Mach. Learn. Appl. 5, 100076. http://dx.doi.org/10.1016/j.mlwa.2021.100076, URL: https://www.sciencedirect.com/science/article/pii/S2666827021000384.

dos Santos, A.A., Junior, J.M., Araújo, M.S., Martini, D.R.D., Tetila, E.C., Siqueira, H.L., Aoki, C., Eltner, A., Matsubara, E.T., Pistori, H., Feitosa, R.Q., Liesenberg, V., Gonçalves, W.N., 2019. Assessment of CNN-based methods for individual tree detection on images captured by RGB cameras attached to UAVs. Sensors 2019 URL: https://doi.org/10.3390/s19163595.

Suharjito, Elwirehardja, G.N., Prayoga, J.S., 2021. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. Comput. Electron. Agric. 188, 106359. http://dx.doi.org/10.1016/j.compag.2021.106359, URL: https://www.sciencedirect.com/science/article/pii/S0168169921003768.

Szegedy, C., Ioffe, S., Vanhoucke, V., 2016. Inception-v4, inception-ResNet and the impact of residual connections on learning. CoRR abs/1602.07261. arXiv:1602.07261. URL: http://arxiv.org/abs/1602.07261.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. CoRR abs/1512.00567.

Tao, C.V., Li, J., 2007. Advances in Mobile Mapping Technology, Vol. 4. CRC Press.

Torabzadeh, H., Leiterer, R., Hueni, A., Schaepman, M.E., Morsdorf, F., 2019. Tree species classification in a temperate mixed forest using a combination of imaging spectroscopy and airborne laser scanning. Agricult. Forest Meteorol. 279, 107744. http://dx.doi.org/10.1016/j.agrformet.2019.107744, URL: https://www.sciencedirect.com/science/article/pii/S0168192319303600.

Xie, Z., Chen, Y., Lu, D., Li, G., Chen, E., 2019. Classification of land cover, forest, and tree species classes with Ziyuan-3 multispectral and stereo data. Remote Sens. 11 (2), 164.

Zhang, Y., Xiao, S., Zhou, G., 2020. User continuance of a green behavior mobile application in China: An empirical study of Ant Forest. J. Cleaner Prod. 242, 118497.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2017. Learning transferable architectures for scalable image recognition. CoRR abs/1707.07012.