Plants recognition using embedded Convolutional Neural Networks on Mobile devices

*Denise Pechebovicz, *Sthefanie Premebida, *Vinicios Soares, *Thiago Camargo,
*Jakson L. Bittencourt, *Virginia Baroncini - *IEEE member* and *Marcella Martins
**Federal University of Technology - Paraná - Ponta Grossa - (UTFPR-PG)*
E-mail: {pechebovicz, sthefanie, viniciossoares, tcamargo, jakson}@alunos.utfpr.edu.br;
{virginia, marcella}@utfpr.edu.br

***Abstract*—In this work we propose a mobile application capable of recognizing Brazilian medicinal plants to be used by universities, students that have not previous contact with the species and professionals working on health centers. We describe the database generation based on the Brazilian Ministry of Health list of medicinal and common toxic plants. We also implement artificial intelligence techniques to perform the recognition task using a class of convolutional neural networks (CNN) focused on lowering the computation resource necessary to run deep learning tasks and also optimizing the execution of the architectures on embedded and mobile devices. The results obtained from the techniques and implemented methods are described and their performance are compared against other neural networks focused on embedded systems.**

***Index Terms*—Image Classification, Convolutional Neural Networks, Embedded Applications, Plant Recognition**

I. Introduction

Plants recognition has demonstrated a wide use in scientific field such as biology, agriculture, medicine and pharmacy [1], and in the future, the recognition network can be implement with traditional medicine information. Usually, a developed application recognizes a plant providing a link with its medicinal information. The medicinal information is an accumulated knowledge which has been widespread from generation to generation, and also has been proved scientifically [2].

The historic figure of the healer has stood the time and the innovations of scientific medicine, lasting to the present day with great demand by the population. This demand is related to the purchasing of medicinal herbs or products made for an alternative medical treatment for health problems, for example medicinal teas. Herbal tea, defined as an infusion drink or cooking leaves, bark or roots, in dry form or fresh plants, contains substances or classes of substances responsible for therapeutic effects [3]. There are differences and peculiarities in each country, within ideas, opinions, values, knowledge, practices and different techniques, which can turn the effectiveness of the scientific prove, in part, influenced by habits, traditions and customs [4], [5].

A major problem related to the use of medicines made by healers is the risk of infection, when using non-potable water in teas, cultivation of plants in contaminated soil, contamination of plants by animals and toxic substances. A possible alternative to this is home plant cultivation, which motivate family farming besides mobilizing the cultivation of a medicinal plant to the production of a herbal medicine.

Also, healers need to learn about the active ingredients of herbs, such as the therapeutic indications and contraindications, effect overdoses, allergic reactions, guiding users about possible medicine chemical interactions or drug poisoning, besides procedures for cleaning, storage and risks involved with a erroneous plant identification.

Besides healers, medical professionals and institutions have been adopted medicinal plants. In 2009, the Unified Health System (in Portuguese, Sistema Único de Saúde - SUS) in Brazil has expanded the list of herbal medicines available in the basic pharmaceutical assistance throughout the country. The Ministry of Health also expects that with the Program, the States may feel encouraged to offer the service with this type of medicine, besides to the 12 Brazilian states that already offer. The Program considers 71 medicinal plants.

There are some plants with active ingredients that promote intoxication in humans and animals, known as toxic plants. The toxic plants include all vegetables that the contact, inhalation or ingestion result in damage to health, both for man and for animals, and may even lead them to death. Many toxic plants are considered ornamental, present in different environments around us, therefore facilitating the risk of poisoning for humans. In addition, the toxins present in plants can directly influence animal production, being capable of promoting serious damage to agribusiness.

According to the data from the National Poison Information Pharmacological System (SINITOX) in 2010 Brazil registered 1,132 cases of poisoning by plants, and 330 in the South part of the country [6]. Therefore, by optimizing the plants recognition, it encourages the study of the plant application and the knowledge for its indications and contraindications. Moreover, recognizing poisonous plants may have great relevance to both animals and human society,improving knowledge of users, seeking to identify their once can identify risks avoiding possible poisoning.

The main objective in this work is to implement a neural network model capable of recognizing Brazilian medicinal plants that can be used by the Unified Health System (SUS) in phytotherapy treatments. The model will be embedded in a mobile device, making the application portable, fast and accurate, providing a powerful hand-held tool.

The tool will be used in universities, by students that have no previous contact with the species and for professionals working on health stations, providing an easy and fast classification avoiding the manipulation of toxic species that can be handled by mistake by a non trained eye.

II. Background

A. Related Works

Artificial Intelligence and neural networks have been proving to be accurate when classifying plants, especially when analysing its leaves [7], classifying 32 kinds of plants with an accuracy over 90%. This task was performed by a probabilistic neural network (PNN) with the purpose of setting up a database for plant protection.

Usually the main goals of applying those techniques to plant classification are related to agriculture, in approaches like classifying crop weeds to build autonomous weeding machines [8], crop disease detection [9] and grain classification [10] but it can be extended to areas like pharmacy, medicine, and botany. The results obtained by [9] are particularly interesting, using a public dataset composed of 54,306 images the developed model was capable of identifying 26 diseases in 14 different crop species, obtaining an accuracy of 99.35%.

B. Convolutional Neural Networks

Convolutional Neural Networks (CNN) became very popular to classify images after Alexnet [11] won the Imagenet challenge (ILSVRC) in 2012 and the tendency followed by this area of artificial intelligence was to create architectures that are more complex, robust and accurate. But this improvements have a cost, to run those architectures a powerful computer usually equipped with a high-end graphics processing unit (GPU) is required.

In this work we use a different approach of CNN, instead of focusing on extremely high accuracy, the architectures used here are focused on latency and efficient implementation on embedded and mobile applications. The networks used are called Mobilenets [12].

Those networks use depth-wise separable convolutions as seen on Figure 2 to make the application lighter, reducing the amount of computational effort to get a prediction. This kind of separable convolution is specially efficient when the output channels of a layer is bigger than input channels. The image convolution is similar to 2D convolution, but it multiplies a pixel's and its neighboring pixels color value by a matrix. Those matrices are usually small (3x3 or 5x5) and called kernels [13]. In regular image convolution each convolution layer uses a kernel with $nxnxc$ dimensions, where n is the the row and column size and c the number channels in the input. In depth-wise separable convolutions first a multiplication by an nxn matrix (depth-wise) is done in each layer and after a multiplication by an $1x1xc$ matrix (point-wise) is applied.

As an example, available at [14], with an input of a $12x12$ RGB image (3 channels), the first convolution layer of the network is a $5x5$ filter and the output of this layer is a $8x8$ 256 channel representation. In order to create the 256 channels the CNN will stack 256 $8x8x1$ representations.

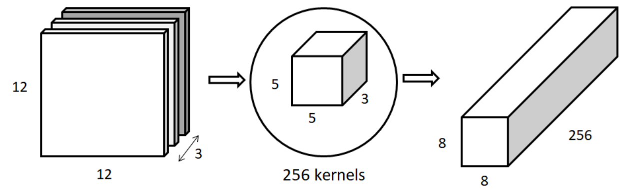
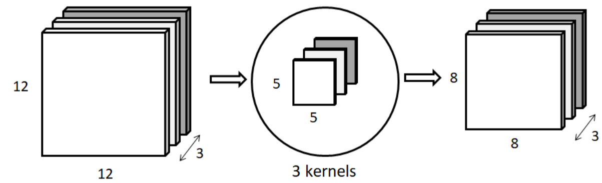
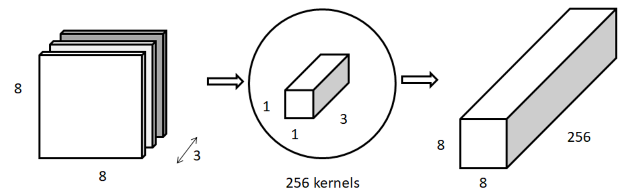


Fig. 1: Normal Convolution



(a) Depth-wise Convolution



(b) Point-wise Convolution

Fig. 2: Depth-wise Separable Convolution: a)Depth-wise Convolution, b)Point-wise Convolution.

- In normal convolution, as seen in Fig. 1, the network will have 256 kernels, with $5x5x3$ dimensions that must move $8x8$ times resulting in 1,228,800 multiplications.
- In depth-wise separable convolutions the network will first execute the depth-wise convolution, as seen in Fig. 2a, using 3 kernels, with $5x5x1$ dimensions that must move $8x8$ times, resulting in 4,800 multiplications. After it will execute the point-wise convolution, as seen in Fig. 2b, using 256 kernels, with $1x1x3$ dimensions that must move $8x8$ times, resulting in 49,152 multiplication. Summing up the two steps the total of multiplications is 53,952.

This method of processing all the image channels separately and rejoining them using a point-wise convolution can make the Mobilenets use from 8 to 9 times less computation than a standard CNN [12].

The Mobilenets have two hyper-parameters that can lower even more the computational cost used to run the application:

- α, a width multiplier that can thin a network uniformly, shrinking each layer. Being $\alpha = 1$ the standard version and the networks with values lower than 1 called reduced Mobilenets.
- ρ, a resolution multiplier that is applied to the input image and the internal representation of every layer, usually this parameter is set implicitly when the resolution is set. The typical value for ρ is 224, but can be as low as 128.

A newer version, Mobilenet v2 [15], improves the state-of-art technology getting better and faster results using inverted residuals and linear bottlenecks. According to the authors of Mobilenet the analysis done in the first version of the architecture provided an empirical hint for optimizing existing neural architectures: as the architecture uses low dimensional layers, assuming the manifold of interest is kept low-dimensional it can be captured by inserting linear bottleneck layers into the convolutional blocks. Inspired by the fact that bottlenecks contains all the necessary information, the architecture uses shortcuts directly between the bottlenecks with the purpose of letting the gradient to propagate across multiple layers. The authors chose to use the inverted residual method as its known to be more memory efficient and showed slightly better results in their experiments.

The architecture of MobileNet v2 starts with a fully convolution layer with 32 filters, the next layers are 19 residual bottleneck and it uses ReLU6 [16], a model where each rectified linear unit consists of only 6 replicated bias-shifted Bernoulli units, instead of an infinite amount, for non-linearity. The basic convolution block can be seen in Fig. 3.

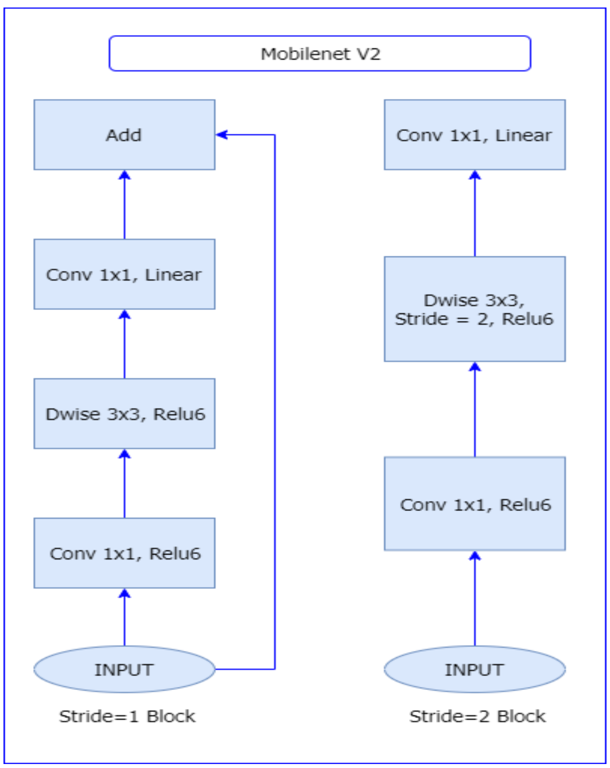


Fig. 3: MobilenetV2 Convolution Block

III. Proposed Approach and Experiments

In this section we present our proposed approach, which considers a network architecture trained to recognize plant images.In our experiments we consider a dataset with 10,162 downloaded images, including different plant life stage, image backgrounds and different environments.

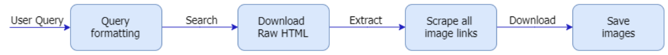


Fig. 4: Image Download workflow

To generate this database, the Google Images Download [17] program was used, allowing the download of images based on keywords or key-phrase. The workflow of the application can be seen on Fig. 4. The standalone version downloads a hundred images per key.

The labels used during the search are based on Brazilian's Ministry of Health [1]. The major labels are plants genus/specie scientific names. To the toxic category we consider a list from Oswaldo Cruz Foundation (FIOCRUZ)[2], a scientific institution for research and development in biological sciences located in Rio de Janeiro, Brazil, considered one of the world's main public health research institutions.

The dataset is composed by 72 categories, the original 71 from the Brazilian's ministry of health plus the toxic category. Fig. 5a and 5b show examples of toxic and nontoxic plant image respectively.

(a) Toxicodendron Radicans (Toxic).

(b) Aloe Vera (Nontoxic).

Fig. 5: Examples of toxic and nontoxic plants

A. Data Augmentation

In this work, the number of training images represents 75% of the original dataset, for each category, as presented in Fig. 6. The category with the highest number of samples is the the one labeled as "toxicas", wich aggregates all the toxic samples in the database, with 1,018 samples in the training set; the one with the lowest number of samples is *Lippia sidoides* with 25.

Despite the increase in sample number, the main goal of augmentation is to improve model generalization generating "new" training data. These techniques are jitters and perturbations. With the new data, the model can train with slightly modified inputs being able to train with more robust features. Usually, applying this method improves accuracy values can be seen.

Data Augmentation task was implemented using the python library Augmentor [18] which includes some augmentation methods, in our data set were used:

- rotation: rotating the images in random angles;

1 Brazilian's Ministry of Health information available at http://bit.do/fa2av

2 The list is available at http://www.fiocruz.br/biosseguranca/Bis/virtual/%20tour/hipertextos/up2/plantas-toxicas.htm

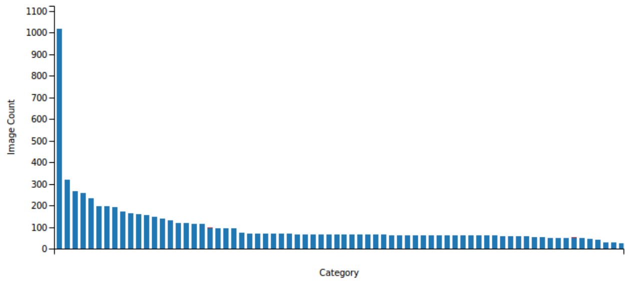


Fig. 6: Original Training Set

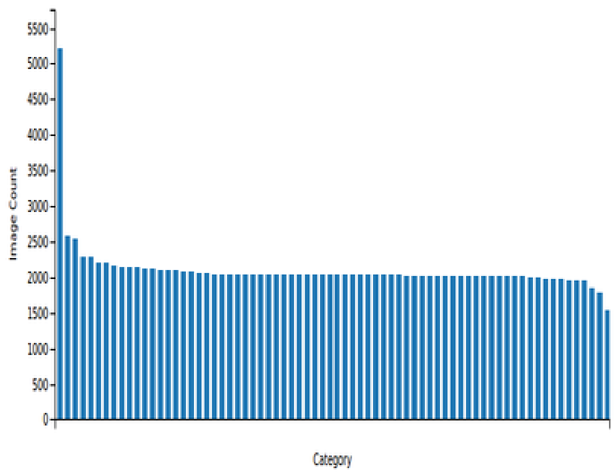


Fig. 7: Augmented Set

- crop: crop the image maintaining the image's aspect ratio;
- mirroring: flipping left side to the right side and vice versa;
- random zooms: applying random zooms in the image;
- random distortions: making distortions to an image while maintaining the image's aspect;
- skewing: gives the impression of looking to the image in a different angle.

After the augmentation as seen in Fig. 8 the total number of pictures were increased to 151,128, with 5,230 in the toxic category and an average number of 2,000 in the other categories as seen in Fig 7.

(a) (b) (c) (d)

Fig. 8: Examples of augmented images: a) original, b), c) and d) augmented

B. Network Architecture

In this work we consider the Mobilenetv2 architecture, using $\alpha = 1$ and $\rho = 224$, with the initial weights trained on the Imagenet dataset. A classifier was added to the model and the convolutional layers were frozen, updating only the classifier weights. The table I shows the model summary where params are the weights applied to the network neurons.

TABLE I: Model Summary

Layer (type)	Output Shape	Param Number
mobilenet v2/1.00/224 (Model)	(7, 7, 1280)	2257984
conv2d (Conv2D)	(5, 5, 32)	368672
dropout (Dropout)	(5, 5, 32)	0
global average pooling2d (Gl)	(32)	0
dense (Dense)	(72)	2376

The complete model has a total of 2,629,032 params, being 371,048 trainable in this step and 2,257,984 non-trainable. To increase the performance a fine tuning is done, training the top layers of the Mobilenetv2 alongside the added classifier. The base model has 155 layers, the layers from 101 to 155 were unfroze and the model are retrained. In this step 2,233,640 params are trainable and 395,392 non-trainable params. Different epoch size are used in both training and fine tuning process (5, 10, 20, 30) and the comparison can be seen in the Results section. The final result is trained for 20 epochs and fine tuned for 20 epochs.

C. Framework

The objective of this work is to use this network model in a mobile device, to do so TensorFlow Lite framework is used, whose tools help to develop neural networks to be applied in mobiles, using TensorFlow models. There are two main components in TensorFlow lite: the TensorFlow Lite interpreter, which can run models in many different hardware, such as mobile phones and micro-controllers; and the TensorFlow Lite converter, which converts a model done in TensorFlow, optimizing the model and enabling it to be embedded. Therefore, the converted network can be deployed to a mobile phone and we can check the latency and the accuracy of the application.

IV. Results

Our dataset is divided into training and validation, using 80% and 20% of the images respectively.The training process using using the original data, without augmentation, and Mobilenet v1 presents poor results. The loss is always higher than 1 and the accuracy is between 0.4 and 0.5. The loss represents how many samples (percentage) the algorithm estimated wrong, and is a number between 0 and 1. For example, a loss of 0.25 represents that the technique gets 3 right answers in 4 samples. The accuracy represents how sure the algorithm is about its estimation. It is a number between 0 and 1. For example, an accuracy indicator of 0.8 means that the technique is 80% sure about its answer. Even when the learning rate is reduced, or the number of epochs increased, the changes in the final indicators are minimal. Increasing the latency of the network does not show significantly increasing of the accuracy either.

The results using Mobilenet v2 are similar with Mobilenet v1, slightly decreasing the loss and increasing the accuracy. To improve the network performance the dataset was augmented and the network trained for 10 epochs then fine tuned for 5 epochs.

As the results seem to be keep improving when increasing the number of epochs, the network are retrained for 30 epochs then fine tuned for 10 epochs. The results can be seen in Figure 9.

To analyse the behavior of the network training, the number of epochs training are reduced to 20 and the fine tuning epochs increased to 20. These results can be seen in Figure 10. The weights obtained from this last training and fine tuning step are used in the embedded model in the mobile phone. It has 0.94 accuracy and 0.18 loss. An accuracy comparison against ImageNet can be seen in Table II.

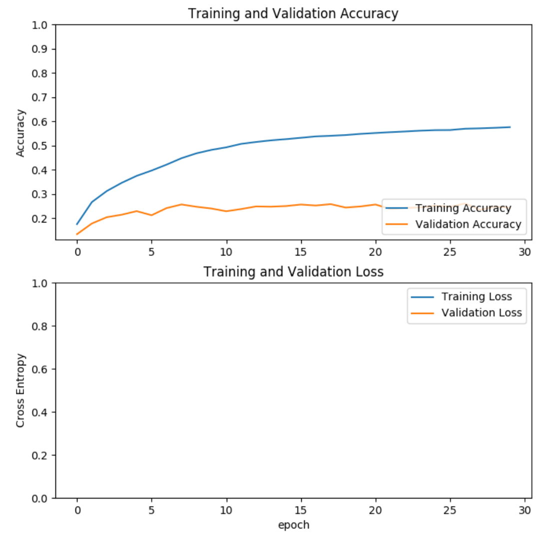
TABLE II: Accuracy Comparison

Network	Accuracy(%)	Dataset
MobileNetV1	70.6 (Top 1)	ImageNet
ShuffleNet (1.5)	71.5 (Top 1)	ImageNet
ShuffleNet (x2)	73.7 (Top 1)	ImageNet
NasNet-A	74.0 (Top 1)	ImageNet
MobileNetV2	72.0 (Top 1)	ImageNet
MobileNetV2 (1.4)	74.7 (Top 1)	ImageNet
MobileNetV2	94	Compiled in this work

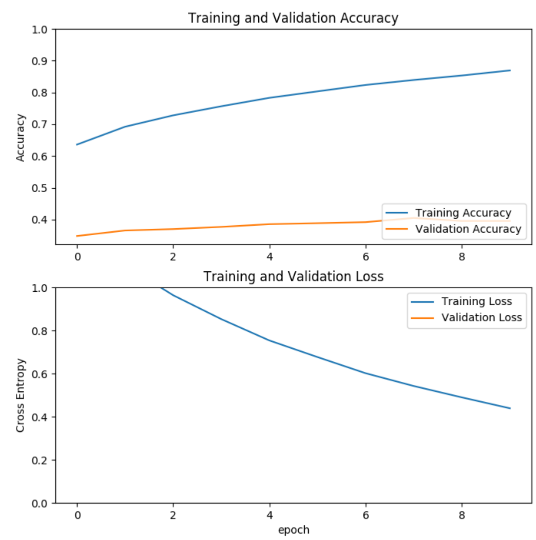
V. Conclusions and Future Work

In this work we implemented a CNN to recognize plants using a mobile device. The network architecture was based on Mobilenet v2 over a database composed by 10162 images downloaded using a Google Images software, being able to classify 72 categories of plants according with Ministry of Health in Brazil. Besides, augmentation methods were applied to increase the generalization of the training set. The framework was developed using TensorFlow, and to fit the network model to a mobile device the TensorFlow Lite Converter tool was applied. Our embedded model presented 0.94 accuracy and 0.18 loss, values significantly higher when compared to other architectures, but as this values were obtained during test, the validation showed that our applications still need some improvement before being used by other researchers.

In future works we plan to expand our database and refine the Google Images Download usage. In addition, we intend to use plants common names as keywords with the grammar accentuation and the letter cedilla. The database may be also increased in images and genus/specie quantity and quality by partnerships with research institutions. Moreover, to enable the user search for more information about the identified plant, we will create a mobile interface able to link the classification to Brazilian's Ministry of Health information and others research institutions.



(a) Training for 30 epochs



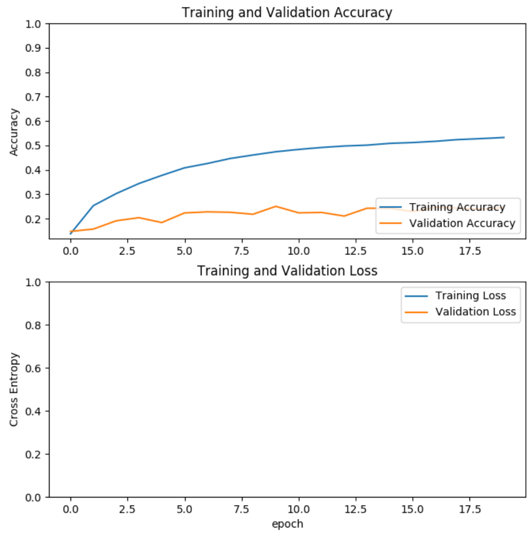
(b) Fine Tuning for 10 epoch

Fig. 9: Second Run: a)Training for 30 epochs and b) Fine Tuning for 10 epochs

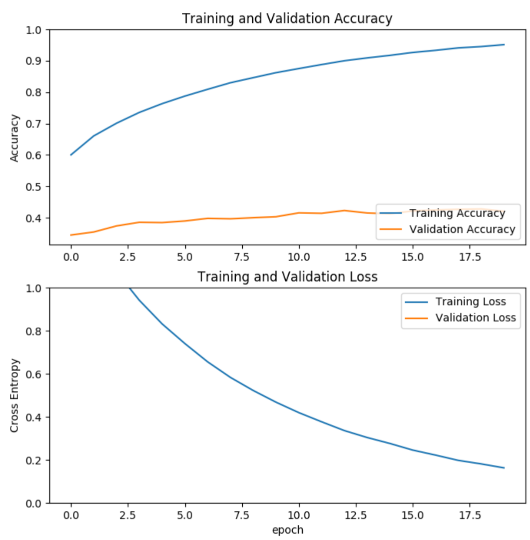
Finally to enhance the neural network performance other augmentation methods can be applied as well as other methods of training and different optimizer, including a performance analysis to better understand the behavior of the network.

Acknowledgements

We gratefully acknowledge the support of UTFPR/Scientific Research program. T. Camargo acknowledges Fundação Araucária.



(a) Training for 20 epochs



(b) Fine Tuning for 20 epochs

Fig. 10: Final Run: a) Training for 20 epochs and b) Fine Tuning for 20 epochs

References

[1] G. D. Antonio, C. D. Tesser, and R. O. Moretti-Pires, "Contributions of medicinal plants to care and health promotion in primary healthcare," *Interface-Comunicação, Saúde, Educação*, vol. 17, no. 46, pp. 615–633, 2013.

[2] V. Schulz, R. Hänsel, and V. E. Tyler, *Rational phytotherapy: a physician's guide to herbal medicine*. Psychology Press, 2001.

[3] F. PEDRO, G. ARRUDA, J. OLIVEIRA, A. SANTOS, K. SIGARINI, T. HERNANDES, R. VILLA, and A. OLIVEIRA, "Composição centesimal e mineral de plantas medicinais comercializadas no mercado do porto de cuiabá, mato grosso, brasil," *Rev. Bras. Pl. Med*, vol. 18, no. 1 supl I, pp. 297–306, 2016.

[4] L. Mathew and S. Babu, "Phytotherapy in india: Transition of tradition to technology," *Current Botany*, 2011.

[5] E. Elisabetsky and L. Wannmacher, "The status of ethnopharmacology in brazil," *Journal of Ethnopharmacology*, vol. 38, no. 2-3, pp. 129–135, 1993.

[6] S. Campos, C. Silva, P. Campana, and V. Almeida, "Toxicidade de espécies vegetais," *Revista Brasileira de Plantas Medicinais*, vol. 18, no. 1, pp. 373–382, 2016.

[7] S. G. Wu, F. S. Bao, E. Y. Xu, Y. Wang, Y. Chang, and Q. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE International Symposium on Signal Processing and Information Technology*, Dec 2007, pp. 11–16.

[8] S. P. Adhikari, H. Yang, and H. Kim, "Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy field," *Frontiers in Plant Science*, vol. 10, p. 1404, 2019.

[9] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 1419, 2016.

[10] N. Visen, J. Paliwal, D. Jayas, and N. White, "Ae—automation and emerging technologies: specialist neural networks for cereal grain classification," *Biosystems Engineering*, vol. 82, no. 2, pp. 151–159, 2002.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, 2012.

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[13] J. Ludwig, "Image convolution," *Portland State University*, 2013.

[14] "A basic introduction to separable convolutions - towards data science," https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728, (Accessed on 10/11/2019).

[15] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[16] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, no. 7, pp. 1–9, 2010.

[17] H. Vasa, "Google images download." [Online]. Available: https://google-images-download.readthedocs.io/en/latest/index.html

[18] M. D. Bloice, C. Stocker, and A. Holzinger, "Augmentor: an image augmentation library for machine learning," *arXiv preprint arXiv:1708.04680*, 2017.