Towards Retraining of Machine Learning Algorithms: An Efficiency Analysis Applied to Smart Agriculture

Jérôme Treboux*, Rolf Ingold †, Dominique Genoud*
*Institute of Information Systems
University of Applied Sciences and Arts Western Switzerland (HES-SO), Sierre, Switzerland
Email: jerome.treboux@hevs.ch, dominique.genoud@hevs.ch
†Department of Informatics
University of Fribourg, Fribourg, Switzerland
Email: rolf.ingold@unifr.ch

***Abstract*—This paper compares the efficiency of state-of-the-art machine learning algorithms used to detect an object in an image. A comparison between a deep learning algorithm such as the VGG-16 and a well-tuned random forest algorithm using classical image analysis parameters is presented. To estimate the efficiency, the classification performances like AUC, precision, recall and computation time of the algorithm retraining process are used. The experimental set-up shows that a well-tuned random forest algorithm is equal to, or better than, the deep learning approach and increases the speed of the retraining process by a factor of around 400.**

***Keywords*-Machine Learning; Decision Tree Ensemble; Random Forest; Prediction; Aerial images; Neural Network; Image recognition; Smart Agriculture; Precision Agriculture; Active Learning; Real-Time Retraining**

I. Introduction

Today, precision agriculture is about increasing and maximizing the productivity of a culture. The entire culture cycle must benefit from the application of the right amount of product to be sprayed at the right time and in the right place (such as water, fertilizers, or pesticides) [1]. Research projects on image analysis for smart agriculture include the detection of plant diseases [2]. Sensors are placed around fields and collect various information. They provide historical information but also real-time information. Recently, drones (UAVs) have improved precision agriculture. These drones are equipped with high-definition cameras to collect aerial images and then map the selected area with photogrammetry software [3] [4]. High-precision agriculture has only recently been applied to complex landscapes and challenging topography.

Machine Learning (ML) algorithms are widely used for image recognition. Deep learning (DL) algorithms, such as the AlexNet Neural Network [5], Convolutional Neural Networks (CNNs) [6], and the VGG-16 neural network [7], are the most powerful for detecting objects in images. Deep Neural Networks are trained on large datasets using the backpropagation algorithm to discover patterns in the image [8]. They represent an image with several levels of abstraction. With the increasing amount of information available, deep learning algorithms are continually improving.

However, in many real-life situations, like high precision agriculture, it is difficult to acquire a dataset important enough to train a Deep Neural Network. This paper is based on a previous research [9], which shows how to tackle this problem using a small dataset. It is based on the use case that aims to increase productivity and maximize crop yields. It focuses on the detection of vines and potential diseases on images taken by a drone (UAV).

This paper compares the performance of two types of pretrained algorithms used to recognize objects in a real-time image. As a baseline, a deep learning (DL) algorithm model trained on our dataset is applied to a picture of a vineyard. Then, a second algorithm called Random Forest (RF) or Decision Tree Ensemble (DTE) model trained on the same dataset is applied to the same new image. In a second step, a retraining process is run on a subset of the new images to evaluate the classification performance and the computational time required for the retraining process. Indeed, the retraining process of a deep learning algorithm is computationally very intensive in terms of time, memory, and GPU/CPU. Other algorithms such as the decision tree have a promising accuracy since 2016 [10][11] and are trained more than 400 times faster than a DL algorithm.

II. State-of-the-Art

Research by Geitgey [12] shows that Deep Learning and CNN algorithms have the best state-of-the-art performance for image recognition. The experiments conducted with Tensorflow [13] have outstanding performance with many output classes.

As also shown in [14], deep models increase the learning performance of classical neural networks. However, deep neural networks require a huge amount of data. For example, the CNN VGG-16 architecture [7] needs to estimate 138 million parameters in total [15].

978-1-7281-6728-2/20/$31.00 © 2020 IEEE

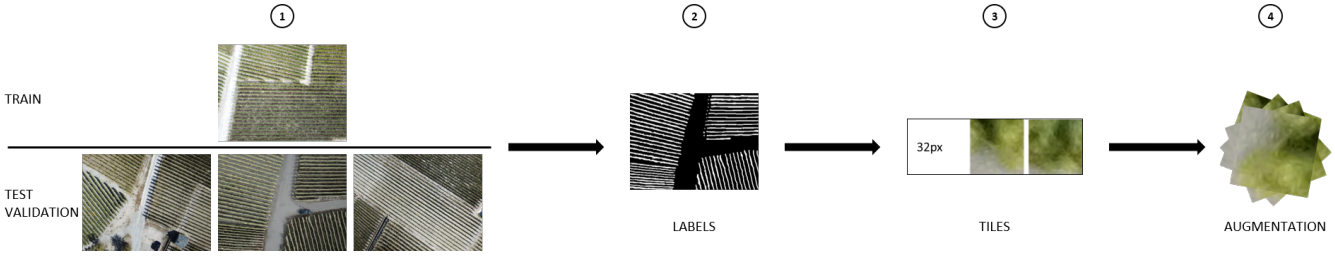

Figure 1. Methodology to create the dataset. (1) The dataset is divided into two partitions. (2) Images are labelled. (3) The images are divided into tiles for classification. (4) The images are augmented.

Others propose experiments using Random Forests for the detection of objects in images, such as face recognition [16]. Besides, their effectiveness has been demonstrated in many areas of the industry [2][17].

There are many other non-ML methodologies for object detection in an image. Their advantages and disadvantages are detailed in [2] and show that ML algorithms are more efficient. Our latest experiments with Random Forests have shown superior performance compared to other machine learning algorithms for the detection of defects in rice grains present in an image [18].

Furthermore, random forests have a better efficiency in terms of computation time during the training process [9]. Indeed, it shows that the computation time to train the Random Forest is much faster than a pretrained VGG-16 on imagenet [19].

Some projects implement a retraining process to increase the quality of their model. It is based on new real-life data. For example, Cisco updates its model in real-time to detect DDoS attacks on the Internet Network [20]. Other projects, such as [21] consist of retraining Neural Networks based on new data acquired with a drone to improve the live detection of objects in videos.

III. Dataset and Data Processing

The dataset is created using a drone flying over vineyards in Switzerland. The drone takes a picture in a high-resolution format. The primary dataset used is detailed in Table I.

Table I
Description of the dataset.

	Settings
Location	Sierre, Valais, Switzerland
Seasons	Summer
Images	790 Aerial images
Geolocalisation	TRUE
Resolution per image	4.000x3.000 pixels
Colour	RGB
Altitude	50 meters

Figure 1 describes the methodology used to create the dataset for the experiments. (1) One image is used for the training partition. The three others are used for the test and

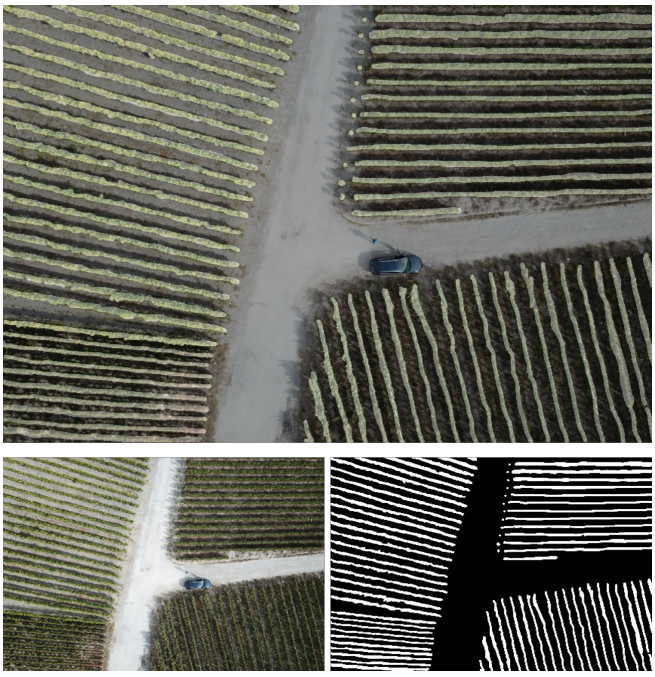

Figure 2. Manually labelled images. Bottom left: original image. Top: original image with the mask containing labels. Bottom right: mask containing labels.

Figure 3. Methodology for the original models creation. (1) From the augmented dataset, (2) the training dataset with the tiles is randomized 5 times for cross-validation. (3) At each step of the cross-validation, multiple training process are executed using an incremental number of tiles from the list. The sampling used is "Take from the top".

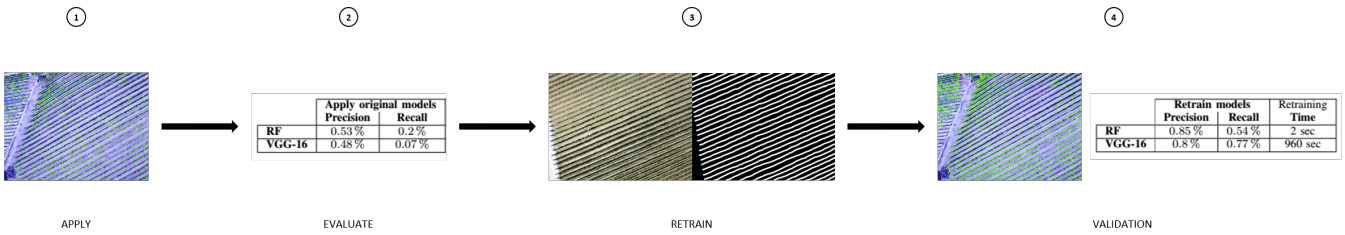

Figure 4. Methodology applied to our experiments. (1) The original models are applied to a new image (validation subset) and (2) the results are evaluated. (3) Then a similar vineyard image is labelled (retrain subset) and used for the retraining process. (4) Finally, the new models are applied to the validation image and the improved results are analyzed.

the validation. The same validation partition is used for all iterations of the experiment. (2) Each image is manually labeled through a mask with white lines representing vine lines (see Figure 2). (3) Based on these labels, the image is divided into tiles. Each tile, depending on its content, is classified as **Vine** or **Other**. A tile that contains more than $33\,\%$ of white pixels is labeled as **Vine**. This threshold is defined in order to obtain the most precise labels and thus not to miss any vine lines [2]. (4) Finally, due to the small amount of data, the lines of the vines do not cover all orientations and therefore do not match the reality of the ground. We perform a dataset augmentation based on the specifications defined by [22].

In the experiments presented in this paper, images of new vineyards were acquired. These vineyards are located in the same region but have different characteristics. For example, the new vines have a distinct geographical orientation, trees are present in the area, and there is more shade due to the timing of the image acquisition. This new dataset is divided into two subsets: validation and retrain. The validation set is used to assess the efficiency of the algorithms and to simulate real-life situations. The second subset is used for the algorithms retraining process to evaluate their new performances and analyze their improvements. The methodology used is detailed in the next section.

IV. Experiment and Settings

The original approach and methodology applied to our experiences are described in this section. We apply pretrained models to a simulated real-time use case. The experiments and their parameters are described as well in this section.

A. Methodology

Figure 3 details the methodology applied to obtain the original and stable models generated in our previous project. The algorithms are cross-validated with five randomized lists. Then, to analyze the evolution of the efficiency of the algorithms, each iteration uses an incremental dataset. The increments are based on "from the top" sampling.

Figure 4 describes the methodology of our experiment. (1) The original algorithms are applied to a new aerial image of a new vineyard (validation subset). (2) The efficiency of the algorithms are evaluated. (3) The second image of a similar vineyard (retrain subset) is manually labeled and used for the retraining process to improve vine line detection. (3) The resulting new models are re-applied to the validation subset and evaluated. The results obtained are presented in the tables III and IV with the Precision, Recall, Computation time, Area Under Curve, and Standard Error.

B. Baseline: Deep Learning - VGG-16

The original deep neural network algorithm is trained with Keras [23], an open-source neural network library using Tensorflow. The pretrained neural network used is the VGG-16 [7], trained on imagenet [19]. This network is used as a baseline.

The network is configured to predict two classes: **Vine** and **Other**. The retraining process and fine-tuning of the VGG-16 is described in [9].

The retraining process is performed as in the original experiment. The last three layers of the VGG-16 are trained using the activation functions **relu** and **sigmoid** with a **dropout**.

C. Random Forest

The training of the RF is done on the following 86 features extracted from the same tiles used for the VGG-16:

First order statistics: Min, max, mean, geometric mean, sum, variance, skewness, kurtosis, etc [24].

Tamura: Granularity, Contrast, Kurtosis of directionality, Standard Deviation Directionality, Max Directionality and Skewness [25].

Haralick: Statistical features based on gray-level co-occurrence matrix [26].

These characteristics are selected based on research carried out by [9]. They represent the patterns and region of interest present in the image. Thanks to the Gini index [27], Random Forest selects the most informative elements internally. The Random Forest configuration is defined in the table II.

Retraining is done from scratch to improve the quality of Random Forest classification. Indeed, a new Forest is created

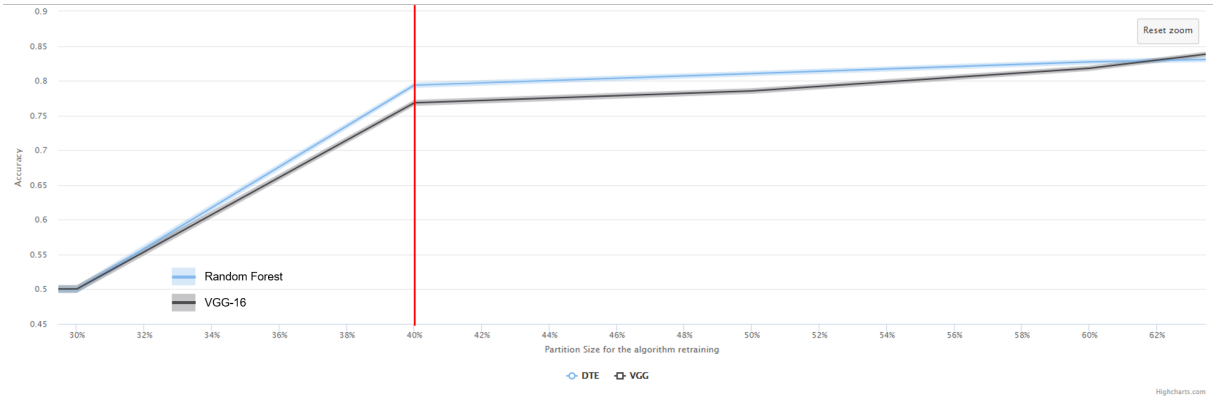

Figure 5. A comparison of the efficiency between Random Forests (DTE) and VGG-16 based on one image of 32x32px for the retrain partition. The X-axis shows the percentage of data used to retrain the algorithms. The Y-axis represents the Area Under Curve. Blue line: Random Forest (DTE) Area Under Curve. Blue surface: The Random Forest Standard Error. Black line: The VGG-16 Area Under Curve. Black surface: The VGG-16 Standard Error. The red line represents the minimum retraining dataset size to obtain a stable classification model.

Table II
RANDOM FOREST CONFIGURATION SUMMARY.

	Settings
Split criterion	Gini Index
Tree depth	No limitation
Minimum child node size	No minimum value
Number of trees	300
Attributes data sampling method	Square root

based on the latest vineyard image (retrain subset), including old aerial images already used for the initial training process. The Random Forest will re-evaluate the characteristics using the Gini index. This process will produce a new Forest with adjusted trees.

V. RESULTS

This section shows the differences in precision and in computing time required to obtain a retrained classification model. The results are presented at the Equal Error Rate to give the same importance to both output classes.

Our results include the Standard Error (SE) calculated based on [28] . The calculation is done with the Equation 1.

$$SE = \sqrt{\frac{\Theta(1-\Theta)+(n_p-1)(\frac{\Theta}{2-\Theta}-\Theta^2)+(n_n-1)(\frac{2\Theta^2}{1+\Theta}-\Theta^2)}{n_p n_n}} \quad (1)$$

Where:

Θ: is the Area Under Curve (AUC)
n_p: is the number of data representing the positive class
n_n: is the number of data representing the negative class

Our results include precision, recall, and computation time for retraining. Precision is described with the equation 2 and recall is detailed with the equation 3.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Where:

TP: is the True Positive
FP: is the False Positive
FN: is the False Negative

First of all, Figure 5 shows that both algorithms require a minimum number of tiles to be correctly retrained. It shows that the Random Forest is significantly more efficient than the VGG-16 with a retrain subset of a size between $40\,\%$ (4650 tiles) and $60\,\%$ (6975 tiles). Table III presents the AUC and the Standard Error with the application of the original and retrained models on the new validation image.

Table III
AREA UNDER CURVE AND STANDARD ERROR OF THE RANDOM FOREST AND VGG-16 ORIGINAL MODELS AND RETRAINED MODELS APPLIED DIRECTLY ON THE VALIDATION IMAGE (SEE FIGURE 5).

	Apply original models		**Retrain models**	
	AUC	**SE**	**AUC**	**SE**
RF	$70.9\,\%$	$\pm 0.5\,\%$	$79.4\,\%$	$\pm 0.8\,\%$
VGG-16	$64.1\,\%$	$\pm 0.5\,\%$	$76.8\,\%$	$\pm 0.9\,\%$

In a second step, the table IV details precision and recall. Precision indicates the number of vine rows correctly classified. Recall indicates the percentage of vine lines that the algorithm correctly identified. Both algorithms have low precision and recall with the application of the original models. This low precision is also visible in the images of the line **ORIGINAL** in Figure 6. The number of correctly

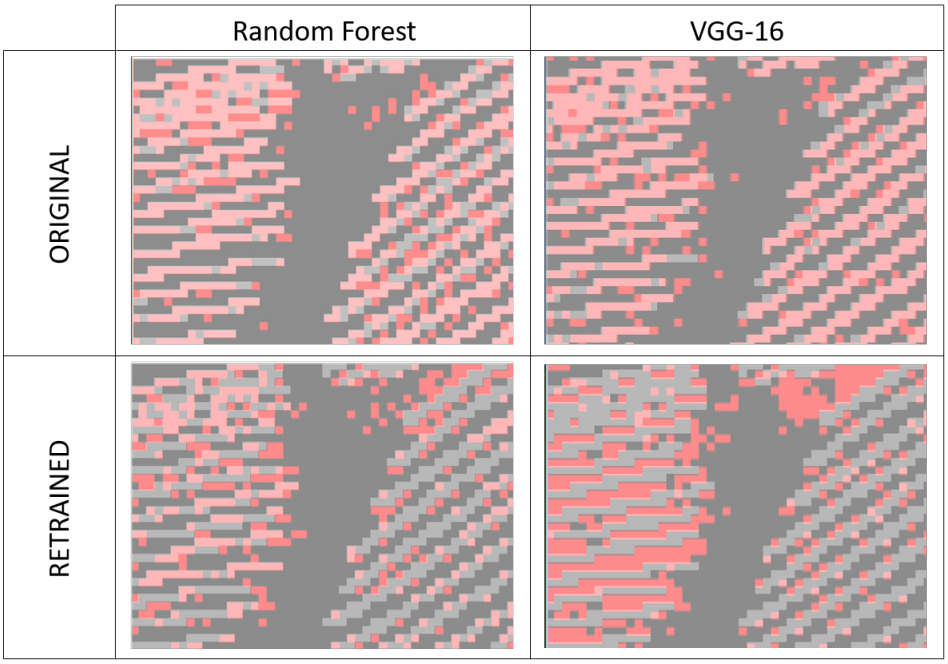

Figure 6. Classification results as presented with our visualization tool. Top left: Classification results with the original model trained with a random forest. It presents tiles *wrongly* classified as **vine** (dark red) or **other** (light red) and tiles *correctly* classified as **vine** (light grey) or **other** (dark grey). Bottom left: the results of the classification with the retraining of the Random Forest. Top right: the results of the classification with the original model trained with a VGG-16. Bottom right: the results of the classification with a retrained VGG-16.

classified vine lines is insufficient. But the recall for RF is still more than twice that of VGG-16.

To visualize the classification results with the original and retrained model, we developed a tool that highlights the tiles wrongly classified in red (See Figure 6). The top line shows the result of the classification when the original models are applied to a new image of a new vineyard. Tiles correctly classified as **Vine** are in light grey, and those classified as **Other** are in dark grey. Tiles wrongly classified as **Vine** are in dark red, and those classified as **Other** are in light red. The bottom line of the figure shows the result of the classification after a retraining process. The lines of vine are generally detected. However, the VGG-16 classifies too many tiles as a **Vine** (dark red). On the other hand, the RF is missing some tiles containing vine (light red).

Table IV
THIS TABLE SHOWS THE PRECISION AND RECALL WHEN THE ORIGINAL MODELS ARE APPLIED TO A NEW IMAGE AND AFTER THE RETRAINING PROCESS. THE TABLE SHOWS THE RESULTS OF THE RETRAINING PROCESS BASED ON THE FULL RETRAIN SUBSET. THE COMPUTING TIME TO RETRAIN A VGG-16 AND A RF IS EVALUATED ON A SERVER WITH 1 TB OF MEMORY (RAM) AND 56 CPUS (2.6 GHZ).

	Apply original models		**Retrain models**		**Retraining**
	Precision	**Recall**	**Precision**	**Recall**	**Time**
RF	53 %	20 %	85 %	54 %	2 sec
VGG-16	48 %	7 %	80 %	77 %	960 sec

With retrained models, precision and recall explode. Random Forest's precision is 5 % higher than VGG-16's. Random Forest's recall is 23 % lower than VGG-16. It shows that the number of vine lines detected is greater with the Neural Net. However, overall detection with a Random Forest is more accurate. In Figure 6, this difference in classification precision is visible.

Finally, the computational time for retraining a random forest is reduced by a significant factor compared to retraining a VGG-16. Both computation times are collected on the same server running on 56 CPUs (2.6 GHz) and 1 TB of memory (RAM).

VI. CONCLUSION

This paper demonstrates that the process of retraining a machine learning model using classical parameters extracted from an image and Random Forest algorithms is similar to a state-of-the-art deep learning baseline (VGG-16-Imagenet). It shows that the random forest retraining process is 400 times faster than the deep learning algorithms.

The Random Forest shows high potential for a real-time retraining process, which can be used for precision agriculture or smart agriculture project, for example. This approach must now be validated with additional classes.

The next step will be to improve results by reducing misclassification. This process will use the visualization tool (see Figure 6) developed to implement an active learning process. This tool highlights the classification errors made by the algorithms. It is used to manually correct mistakes

to enhance the training dataset and the resulting Machine Learning models.

ACKNOWLEDGEMENT

The authors would like to give special thanks to Justin Capick from School 42 in Paris for the development of the visualization and active learning tool. This tool is used for all our internal project to analyze the efficiencies of the algorithms and integrate the human expertise in the Active Learning process.

REFERENCES

[1] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming–a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.

[2] J. Treboux, D. Genoud, and R. Ingold, "Decision tree ensemble vs. nn deep learning: efficiency comparison for a small image dataset," in *2018 International Workshop on Big Data and Information Security (IWBIS)*, pp. 25–30, IEEE, 2018.

[3] C. Anderson, "Agricultural drones," 2014.

[4] Aero41, "The most advanced spraying drone on the market!." Online.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[6] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

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

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, may 2015.

[9] J. Treboux, R. Ingold, and D. Genoud, "Drastically improving the training speed of agricultural object recognition using small training datasets." To be published.

[10] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[11] A. Gonzalez, D. Vazquez, A. Lopez, and J. Amores, "On-board object detection: Multicue, multimodal, and multiview random forest of local experts," *IEEE transactions on cybernetics*, vol. 47, no. 11, pp. 3980–3990, 2016.

[12] A. Geitgey, "Deep learning and convolutional neural networks." [Online] Medium.com, June 2016.

[13] Google, "Tensorflow." [Online] https://tensorflow.org.

[14] R. García-Ródenas, L. J. Linares, and J. A. López-Gómez, "On the performance of classic and deep neural models in image recognition," in *Artificial Neural Networks and Machine Learning – ICANN 2017*, pp. 600–608, Springer International Publishing, 2017.

[15] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, pp. 117–122, IEEE, 2018.

[16] E. Kremic and A. Subasi, "Performance of random forest and svm in face recognition.," *Int. Arab J. Inf. Technol.*, vol. 13, no. 2, pp. 287–293, 2016.

[17] M. V. Moreno, L. Dufour, A. F. Skarmeta, A. J. Jara, D. Genoud, B. Ladevie, and J.-J. Bezian, "Big data: the key to energy efficiency in smart buildings," *Soft Computing*, vol. 20, no. 5, pp. 1749–1762, 2016.

[18] C. Mayoraz, J. Treboux, M. Graeber, and D. Genoud, *Industrie 4.0: The Shapers*, ch. Computer Vision: reconnaissance automatique de grains de riz défectueux par machine learning, pp. 72–75. Georg Editeur, 2019.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[20] K. T. Reddy, D. G. Wing, B. H. Anderson, and D. McGrew, "Automatic retraining of machine learning models to detect ddos attacks," Jan. 4 2018. US Patent App. 15/245,886.

[21] Q. Wu and Y. Zhou, "Real-time object detection based on unmanned aerial vehicle," in *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, may 2019.

[22] V. Schülé, "D´etection automatique d'objets agricoles avec du machine learning," Master's thesis, MSc in Business Administration, University of Applied Sciences and Arts Western Switzerland (HES-SO), 2018.

[23] A. Gulli and S. Pal, *Deep Learning with Keras: Implementing deep learning models and neural networks with the power of Python*. Packt Publishing, 2017.

[24] J. Lu, G. Wang, and P. Moulin, "Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 329–336, 2013.

[25] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, man, and cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[26] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[27] R. Dorfman, "A formula for the gini coefficient," *The review of economics and statistics*, pp. 146–149, 1979.

[28] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.