International Journal of Intelligent Information Technologies
Volume 18 • Issue 2

A New Hybrid Model of Deep Learning ResNeXt-SVM for Weed Detection:
Case Study

Brahim Jabir, LIMATI Laboratory, Sultan Moulay Slimane University, Morocco*
https://orcid.org/0000-0002-8762-9199

Noureddine Falih, LIMATI Laboratory, Sultan Moulay Slimane University, Morocco

ABSTRACT

A set of experiments has shown that deep learning as well as traditional learning can be used in the weed detection process and perform well, although sometimes these models cannot fully exploit and utilize the long-term dependency relationship between some key features of images and image labels. To remedy this known problem in the field of image classification, the authors have introduced a classifier known as the linear support vector machine (SVM). Specifically, they have combined a ResNeXt and SVM network to provide the ResNeXt-SVM framework that can deepen the exploitation of the structured features of images and the understanding of their content. The experimental results show that compared to other algorithm models such as ResNet, ResNeXt, and VGG, the proposed solution is more precise and efficient in classifying weeds.

KEYWORDS

CNN, Deep Learning, Digital Agriculture, Machine Learning, ResNeXt, SVM, Weed Detection

INTRODUCTION

Weed control in fields involves identifying and characterizing the type of weed. As such, the automatic programmed method allows the recognition and classification of known herb types have important applications in this field. Based on the importance of classifying weeds in controlling weeds, researchers have proposed numerous algorithms to identify cultivated weeds (Pulido et al., 2017) (Forero et al., 2018). Researchers classified weeds using SVM with a recognition rate of 98% (Dos Santos et al., 2017). Others have used images to replicate the same experiments with other lenses. They implemented the automatic threshold and adaptive contour to segment parts, and used the method known as the smallest error for classifying the classes, and the recognition rate was on average 96% (Olsen et al., 2015). Other experiments used other algorithms based on K-nearest neighbors. However, this algorithm does not handle samples well, which are sometimes or always unbalanced. If the sampling capacity of one class is higher, while the sampling capacity of other

DOI: 10.4018/IJIIT.296269

*Corresponding Author

classes is lower, it causes problems. To understand this concept well, when a sample that is new enters the system to be diagnosed, it can generate a class with a great capacity to be dominant in the K nearest neighbors of this new sample. Another influencing circumstance is that this algorithm is computationally expensive due to the categorization of each sample in order to calculate the distance of the sample from all other known samples in order to obtain its K nearest neighbors. Nevertheless, from another point of view, this algorithm is more adequate for larger data samples, and for data with few samples, this approach is more likely to may produce misclassified classes. Whether the result is interesting or not, these experiments only use a limited quantity of images, or use their database with modifications made to test their identification, after these manipulations the results become biased (Muppala & Guruviah, 2020). To our knowledge, there are no large public databases for the detection and classification of weeds that have spread in the environment of our study. Therefore, in order to identify the real effectiveness of the proposed algorithm objectively, we collected a set of real field images taken by our professional camera and combined them with images that we obtained on from public dataset released with free licenses. This dataset is filtered and preprocessed, to generate over 3000 images (comprising a large portion of 75% training data and 25% test data). In these datasets, we divide the weeds into 4 classes according to the types of these weeds, which are; Beta vulgaris subsp, Capsella, Chenopodium, Galium aparine (Jabir et al., 2021). These weeds are the most common in the study environment.

With the development of characteristics of computers, which improve its performance, the convolutional neural network has become more powerful and popular in different problems; various improvements of the basic CNN architecture have been made from the nineties to the present day. These changes for the purpose of improvement can be distinguished between regularization, optimization of parameters, structural reformulation, etc. However, it can be said that the goal of improving the performance of CNN comes from reformulating and processing units and building new blocks. Most of the improvements in architecture of CNN have been made on depth and space. The types of innovations made architecture can be classified into seven different categories, namely; spatial exploitation, depth, multipath, width, characteristic map exploitation, channel amplification, etc (Milioto et al., 2018), among the CNN models which interest us a lot in this weed detection experiment, we cite, ResNet (Mahajan & Chaudhary 2019), VGG and ResNeXt (Hitawala, 2018).

The problematic addressed by this article revolves around the problem of image segmentation of a weed, the segmentation rate offered by existing models differs from one study to another but in general it is difficult because it is a question of detecting sometimes very unlike objects on different backgrounds, with the risk of confusing several distinct objects with a single object. In this article we reviewed the existing methods to improve the performance of the prediction obtained by the machine learning models, then we proposed a solution that meets the requirements of machine learning which is an improved algorithm based on the model ResNeXt and SVM, this method has not yet been proposed in the literature review. The last section analyzes and discusses our approach. To conclude with a summary of the novelty and usefulness of our proposed approach and suggests improvements and implementations for ultimate work.

BACKGROUND

Machine learning algorithms allow us to build a predictive model from historical data and use it to predict new data. From the advent of this technology until today, scientists have tried to find ways to develop the model and find solutions to get better machine learning predictions. Many methods have emerged to improve deep learning to achieve better results, in the next section we will try to talk about them. We will divide them into four families, namely: data changes, Machine Learning algorithm methods, model settings, set methods.

Improving Performance With Data

There is no deep learning without the presence of data, so we can say that data is the fuel of deep learning with which it achieves the desired results, and the first process to be considered for to get better results is to pay attention to the data as this gives the biggest performance gains. The ideas of data processing are related to collecting more data, producing new data, purifying and filtering data, transforming data and, the following part is a detail of these processes (Sun et al., 2017).

Collect More Data

Most scientific research has shown that the results of deep learning and other modern nonlinear machine learning techniques increase with more data (Nakkiran et al., 2019). The amount of data is one of the main criteria that makes deep learning so interesting, as shown in Figure 1 below, the more training data a model has, the better it can predict good results.

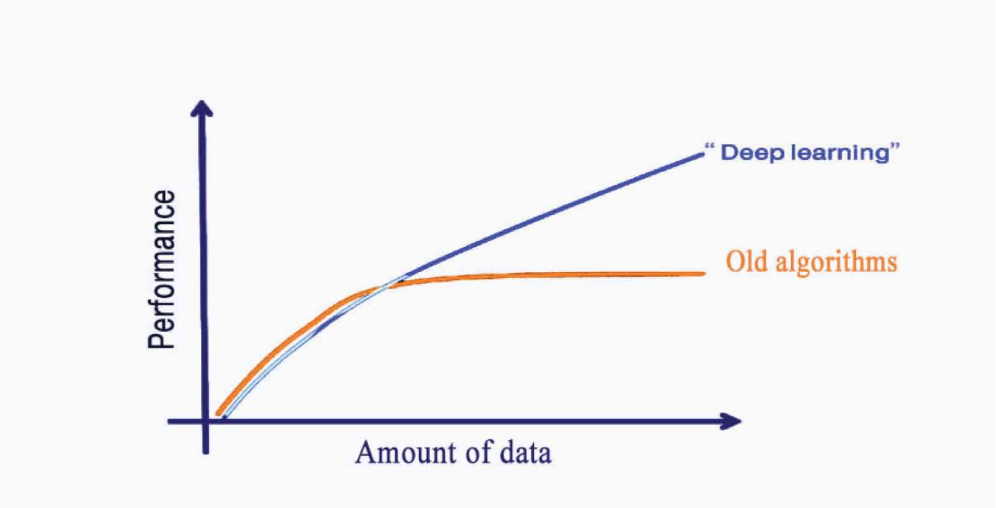
Although more data is important in the machine learning process, it does not always lead to better results, sometimes this can be the cause of overfitting (Rice et al., 2020), so we must take into account the question of quantity and quality at the same time.

Generate New Data

As we saw in the previous part, the amount of data is an important factor to achieve better performance. However, in many areas, this task is difficult; it is not always easy to get more data for several reasons. So there is an alternative solution, which is to produce more data using specific methods, this data is statistically close to the original data, and then a lot of data can be obtained and training can be done on it (Van Tulder & de Bruijne, 2015). Too many synthetic data production techniques differ according to the type of data. If, for example, they are images, some methods can be used like generating randomly modified versions of existing images, if the data is vectors of numbers; from existing vectors, it is highly possible to create other improved versions at random. In addition, if the data is text, other methods can be used. The methods that can be used are as follows:

- The increase of existing data;
- Use a probabilistic model: Monte Carlo method;

Figure 1. The evolution of DL performance with the amount of data



- Use a Deep Learning technique: Variational Autoencoder (VAE) or Generative adversarial network (GAN) (Mahajan & Chaudhary 2019).

Clean Data

Data cleansing is the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data that may negatively affect a predictive model, and then modify, replace or delete it as necessary. Data cleansing is considered a fundamental part of machine learning. There are many types of errors in a dataset, although some of the simpler errors include columns that do not contain much information and duplicate rows (Chuck et al., 2017).

Resample Data

A resampling method is a tool of repeatedly drawing samples from a set of data and calculating statistics and metrics on each of those samples in order to obtain more information. This step is very important in case the data is unbalanced. Over-sampling or under-sampling are techniques to best represent the dataset (Johnson & Khoshgoftaar, 2020).

What makes the use of this resampling method necessary is "random chance". Whenever we take a sample from a population and calculate a statistic, the estimated parameter may be this close to the true value of the population, only due to chance, so we have no information on the confidence of our conclusions and our prediction will be doubtful (Osband et al., 2018). This is why we use "bootstrapping", by taking several samples from our original sample (resampling), we can explore the different combinations that could result in the first place, by calculating standard errors and confidence intervals, which gives us a consistent range of values to estimate the true population parameter.

Features Scaling

Features scaling is a method many in the field of deep learning used to normalize the range of independent variables or characteristics of data. In computing, this is usually done during the data-preprocessing phase (Thara et al., 2019). If we have several independent variables such as age, salary and height; with their range of (18 to 80 years old), (20,000 to 65,000 dirhams) and (0 to 5.5 meters) respectively, scaling the features would help them all be in the same range, for example centered around 0 or in the range (0.1) depending on the technical scale. The two most common scaling techniques are standardization and normalization (Huang et al., 2019) (Pal & Sudeep, 2016), these techniques are used to raise the performance level of algorithms that use weighted inputs.

Features Selection

Features selection is the process that automatically or manually selects the features that contribute the most to the predictor variable. Having irrelevant characteristics in the data can decrease the accuracy of machine learning models (Shi et al., 2018). The common methods of "features selection" are (Doreswamy & Nigus, 2020):

- Univariate Selection;
- Feature Importance;
- Correlation Matrix Heatmap.

Features Engineering

Features engineering refers to the process of selecting and transforming variables/characteristics in the data set when building a predictive model using machine learning. Therefore, functionality must be extracted from the collected raw data set before training the data into machine learning algorithms. Otherwise, it will be difficult to get good information about your data (Zhong et al., 2020).

This method has two purposes. The first is to prepare the appropriate input data set compatible with the requirements of the machine-learning algorithm. The second is to improve the performance of machine learning models.

Data Transformation

Machine learning algorithms such as linear regression and Gaussian Naive Bayes assume that numeric variables have a Gaussian probability distribution.

The data may not have a Gaussian distribution and instead have a Gaussian-like distribution (e.g. almost Gaussian but with outliers or skewness) or an entirely different distribution (e.g. exponential) (Schreiber, 2017).

As such, it is possible to achieve better performance over a wide range of machine learning algorithms by transforming the input and/or output variables to have a Gaussian or more Gaussian distribution. Power transformations such as the Box-Cox transformation and the Yeo-Johnson transformation provide an automatic way to perform these transformations on data and are provided in the Python machine learning library Scikit-learn (Hao & Ho, 2019).

Clustering

Clustering is considered among the best-known and most important unsupervised learning problems; thus, like any other problem of this type, it works with uniquity data in which it tries to find a structure. A definition taken from the literature is “the process of organizing objects into groups whose members are similar in some way or another”. A cluster is therefore a group of objects “similar” to each other and “different” from objects belonging to other clusters (Kao et al., 2008).

The purpose of clustering is to determine internal clustering in an unlabeled dataset. We can assure that there is no better criterion that would be independent of the final objective of the regrouping. Therefore, the user must provide this criterion, so that the result of the grouping meets the specified objective.

Improve Performance With Machine Learning Algorithms

There are several possible ML algorithms and several “Machine Learning models” the choice of the right machine learning algorithm depends on many factors such as the problem being addressed and the type of output you want, the type and size of the data, the time to calculation available, number of attributes, calculation and storage resources, etc. To develop a predictive model after choosing the appropriate context, it is necessary to try a number of models until you find the right model that gives good results (Steiger, 1990). There are also several techniques that can be applied to the algorithm.

Resampling Method

This resampling method is used in a way necessary to exploit the training data in an efficient manner. He considers learning the parameters of a prediction function on data and doing the test on the same data as a methodological error: in another way we can say that the model which would repeat only the labels of the samples that it has already seen would still have perfect precision but really would not predict anything useful on new data (Browne, 2020). Cross-validation is the most efficient technique for performing this task. The data of which is divided into sets for learning, validation and testing. To estimate model performance, we often use some of the data for training and keep some of it for testing, so model performance on test data is representative (Li et al., 2020).

Evaluation Metric

Evaluating the performance of the ML algorithm is an important method of any Deep Learning project. We cannot tell whether a model is performing well or not if we do not evaluate it using one of the separate evaluation metrics that best captures the requirements of the problem. The model can give reliable results when judged using a metric, cited here; *Accuracy_score*, but may give poor results when judged against other metrics like *Logarithmic_loss* or any such metric. Most of the time, we use the accuracy of the classification to measure the performance of our model, but this is not enough to really judge the model (Jabir et al., 2021). Here are some metrics that can be used to judge a ML model:

- Accuracy classification;
- Logarithmic loss;
- Confusion Matrix;
- Area under Curve;
- Mean Absolute Error;
- Mean Squared Error;
- F1 score.

Benchmark Performance

One of the methods to better evaluate and compare ML algorithms for machine learning is the Baseline, it relies on creating a baseline performance on a predictive modeling problem defined as a simple model used as a point reference. This approach offers a comparison tool for advanced methods subject to further evaluation (Choudhary & Gianey, 2017). The two most commonly used Baseline algorithms are the random algorithm and the zero rule algorithm.

Test Some Linear and Non-linear Algorithms

These types of algorithms are the best known and well understood in statistics and machine learning, easy to build and train quickly. In the event that these linear algorithms give important predictive results, they should be preferred because they are easier to improve or modify. The process of evaluation is always important to determine the most efficient linear algorithm and to improve it if they are weak (Pavlyshenko, 2016).

For the non-linear algorithm, it needs many data to work well and give good results; it is more complex but gives good results in most cases. In addition, its evaluation is an essential element to know the degree of learning and to make improvements if necessary (Ouyang et al., 2019).

Standard Configuration

Every algorithm, regardless of its type, needs basic parameters to start learning and predicting. At this point, nothing is changed in the configuration of the algorithm, but we need to know the parameters so that we can change them for better results and be able to solve the problem.

Improve Performance Through Model Parameters

The different deep learning algorithms are distinguished by parameters that generally influence the prediction and performance of learning. The general objective of configuring these parameters is to test the value for each parameter in order to turn the optimum value in order to improve the accuracy of the model (Ranjan et al., 2018). For the tuning of these parameters, we need to have a good understanding of their individual impact on the model set, thus understanding its meanings. Moreover, for that it is necessary to repeat with a certain number of models in order to find the successful model.

As an example: in a “random forest” algorithm, we have a set of necessary parameters like max_features, number_trees, random_state, oob_score and others (Narendran et al., 2021). Finding the most efficient values and the intuitive optimization of those parameter values will result in better and more accurate models. For CNN type models, the known parameters to adjust are filters, size, number, pooling, number of layers of neurons, number of neurons per layer... (O’Shea et al., 2018).

Use the Configuration Cited in the Literature

The easiest and most efficient way to start setting up a machine-learning model is to rely on parameters cited in the literature, which have already been tested and proven effective for a particular problem. These parameters can therefore be used as a starting point to configure a powerful model (Zhou et al., 2020) (Hoiem et al., 2021).

Optimization of Hyperparameters

Deep learning models have hyperparameters. These hyperparameters are configuration points that allow you to configure a training model for a specific task using specific data. There is always confusion between parameters and hyperparameters, hyperparameters are to be configured and set manually to obtain reliable results while the parameters are learned automatically.

The right configuration requires the right maintenance of the hyperparameters, which give the best results of a training model. This is reflected in the search for hyperparameters, the adjustment of hyperparameters, or the optimization of hyperparameters (Reimers & Gurevych, 2017). There are two optimization methods which are used a lot and which are simple, namely the search by grid and the random search. Grid search. Define hyperparameters in the form of a grid in a search space and rate each location in the grid. Random search defines a delimited domain of hyperparameter values as a search space and randomly samples points in that location.

Improving Performance Using Combination Methods

One of the methods used to improve the performance of machine learning models is to combine a set of predictions from multiple ML models. So different models can be combined, each with different results on a problem, and each model can be specialized in a particular process or in a particular part of machine learning (Kotsiantis et al., 2006). The known methods in this process are:

- Merge predictions from several models (Jahrer et al., 2010);
- Merge the predictions obtained on different data representations (Kotsiantis et al., 2007);
- Bagging (Mi et al., 2019) (Zhang et al., 2021) (Hothorn & Lausen, 2003);
- Stacking (Deng et al., 2012).

Proposed Model: A New Hybrid Model

Among the means of improving performance that we have discussed above, in this part we will try to combine a set of these methods and come up with a new hybrid model that can be used in the field of agriculture.

The method we used is the merge method, where a group of models can be combined, each model has been tested with the same weed detection problem, and they have different results as shown in literature. We have also carried out a number of operations related to improving the performance of the combined model, especially, the means related to the parameterization of the model and the preparation of data, diagnostics of the algorithm, etc.

Related Works

In the literature review, a significant amount of research using deep learning methods has considered optimal results in certain tasks related to different problems, using different methods: Natural language processing, speech recognition, text classification, and classification of images (Li et al., 2019) (Deng & Liu, 2018) (Nassif et al., 2019) (Kowsari et al., 2017). The models used in said tasks use the Softmax function at the classification layer. However, studies (Agarap, 2018) (Alalshekmubarak & Smith, 2013) have developed more improved solutions based on SVM, which is combined with another model and each part has a specific task, which gave us the idea to look for similar and more efficient solutions. These explored articles confirm that the choice of the classification method is an important criterion influencing the results, and that the use of the conventional Softmax function in a neural network architecture (ANN) is less reliable compared to models, which uses SVM. Of course, it is known that SVM tries to determine the optimal hyperplane between two classes, then these results in ignoring a multi-nominal case; the positive class represents the class with the highest result while the rest represents the negative class. This is expressed as, the restriction to binary type classification, and it is among the drawbacks of methods using SVM. The applications of deep learning in agriculture are divided into several areas, the most popular being weed identification, land cover classification, plant recognition, fruit counting, and crops type classification. Deep learning models have multiplied and diversified in crop planting, and CNN (Convolutional Neural Networks) have taken the largest share, its derived models such as ResNeXt have been used in this field and have proven to be useful (Kim et al., 2021). SVM is also present in these processes related to weed detection with good results. In examining the data sources used to train the learning model for each study done in this area, they mainly used large image datasets, in some cases containing thousands of images. Some of these images and datasets come from well-known and publicly available sources such as PlantVillage, LifeCLEF, MalayaKew and UC Merced (Kamilaris & Prenafeta-Boldú, 2018).

As shown in the previous sections, most of the research has used CNN architectures to create deep learning models that detect weeds. In addition, they compared them to other models in terms of accuracy and error. Further benchmarks have shown that there is still a need for a more robust model in terms of learning time and accuracy. All the information below and the results obtained show the reasons for the choice and why these two models (ResNeXt-SVM) were chosen to assemble them, and explain the approach that we have proposed as an alternative to traditional methods, which was able to achieve higher results. The following sections explain the details of this solution.

Methods and Materials

In this study, a set of deep learning techniques are used, we used the libraries of Keras and Tensorflow for Python to build and train the model, we also used the Tensorboard charts to diagnose and evaluate the performance, and we used a data set for conducting training (Vogelsang & Erickson, 2020). We will discuss this approach, which combines ResNeXt and SVM in order to achieve a better result.

Dataset

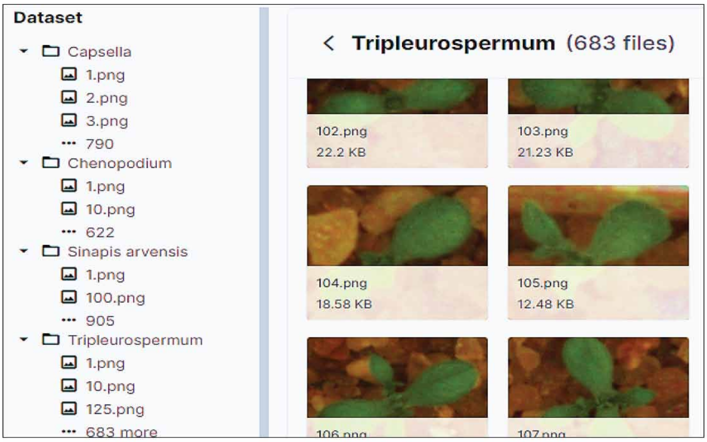
There is no deep learning without the presence of data, and data is one of the direct reasons for whether or not learning is successful. In addition, if we have a large and well-prepared dataset, we will achieve improved training accuracy (Sun et al., 2017). The dataset used in this experiment is a dataset that we previously used in an object detection study that identifies weeds using CNN with some additions and improvements (Figure 2). The images contained are a mixture of images that we took in the fields with a professional camera and a collection of public images that we downloaded from the Internet and synthetic images. The image annotation is performed by Kili technology.

It contains 2000 images of four types of weeds; these images are collected using a professional camera in agricultural fields. This has provided us with a large dataset that we can put online as a public database for the data science community used to achieve data science goals. We have made improvements to these pictures (quality, resizing, contrast, mosaic…). We also applied augmentation techniques (cropping, rotating, flipping...) to obtain a fan of almost 3000 images. It is true that this number of images is still a small number, according to a scientific report (Olsen et al., 2019) Olsen confirmed that we need 17509 images to get more accurate results, but this number that we have will help as a first stage to discover the ability of our model, and we will do another work by producing the most number of images.

The Support Vector Machine (SVM)

The support vector machine (SVM) appeared thanks to the researcher Vapnik (Vapnik et al., 1995) binary type classification problems. Its main goal is to identify the optimal hyperplane $f(w, x) = w.x$

Figure 2. The dataset used in our experiment



+ *b* to put a separation of two classes in a precise dataset, with details and characteristics $x \in R^m$, *w* are parameters learned by SVM by solving an optimization problem as given in equation 1:

$$\min \frac{1}{p} w^t w + c\sum_{i=1}^{p} \max\left(0,1 - y_i^{'}\left(w^T x_i + b\right)\right) \tag{1}$$

where w^T and w called the L1 norm more commonly referred to as the Manhattan norm, C is a value selected using hyperparameter optimization and improvement, it can also be an arbitrary value, it is called the penalty parameter, y' is the real label of the study data, and $w^T x + b$ is the function of prediction. Equation 1 known in the scientific community by the expression: L1-SVM, with the loss of standard hinge. Its derivable counterpart, L2-SVM (equation 2), provides results known by increased stability (Vapnik et al., 1995):

$$\min \frac{1}{p} \|w\|_2^2 + c\sum_{i=1}^{p} \max\left(0,1 - y_i^{'}\left(w^T x_i + b\right)\right)^2 \tag{2}$$

where w_2 is the Euclidean norm (also called the L2 norm), with the hinge loss squared.

ResNeXt Neural Network

ResNeXt inherited from ResNet, VGG and Inception, proposed by Xie (Xie et al., 2017), ResNeXt includes shortcuts from the previous block to the next block, stacking layers and adapting the split-transform-merge strategy. To fully understand how Resnext works, we need to know how ResNet, VGG, and Inception work. ResNet is characterized by the introduction of a shortcut from the previous layer to the next layer, VGG: taking advantage of repeated layers to create a deep architecture model, Inception is based on the principle of "split-transform-merge" to split the entry into multiple blocks and merge blocks later. We conclude from all this that the principle of ResNeXt is to stack the same topology blocks.

Within the residual block, the hyper-parameters are shared. ResNeXt is known under a basic architecture, which is defined by the rules: if the blocks produce spatial maps of the same criteria and dimension, they generalize the same set of hyperparameters, and if the spatial map is down-sampled by a factor of 2, the width of the block is multiplied by a factor of 2. This makes it structured with the building block shown in the figure 3.

The cardinality of this model is placed by the author, which reflects the size of the set of transformations. As shown in the following figure 4, the architecture consists of 32 blocks of these blocks have identical topologies, the cardinality has a value of 32. Due to the use of the same topology, fewer parameters are requested while other layers are added to this architecture.

There are many variations of the ResNeXt architecture, that means the same concept but with different number of layers and different parameters. We are interested in ResNeXt50 and ResNeXt101, ImageNet has been used to show the improvement in accuracy when cardinality is taken into account rather than width/depth, the following Figure 5 compares ResNeXt50 and ResNeXt 101 with different parameters (Bansal et al., 2018). ResNeXt-50 and ResNeXt-101 have low errors when the cardinality

Figure 3. Left: Architecture of ResNet and right: Architecture of ResNeXt (Xie et al., 2017)

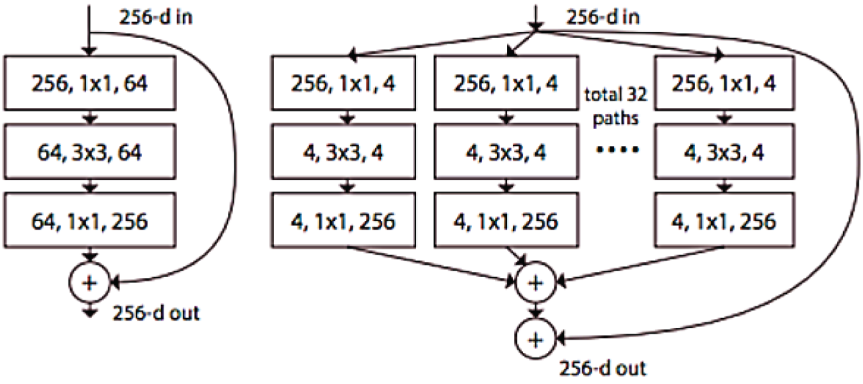
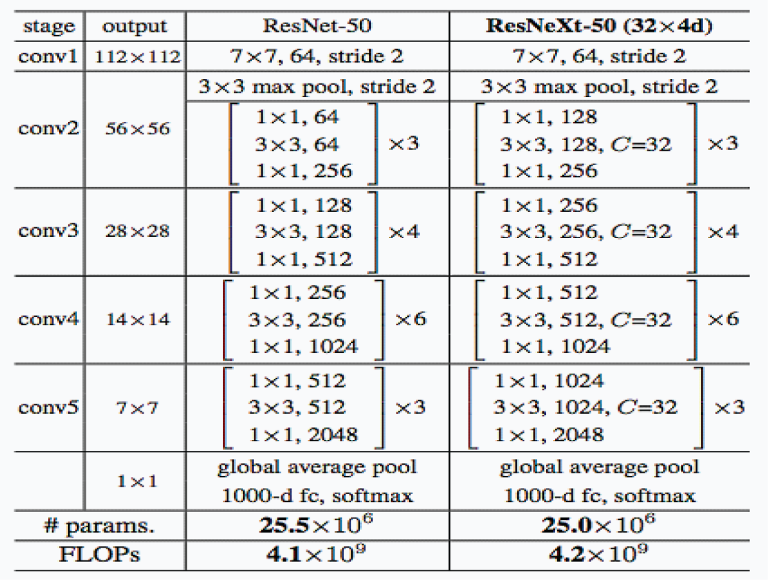
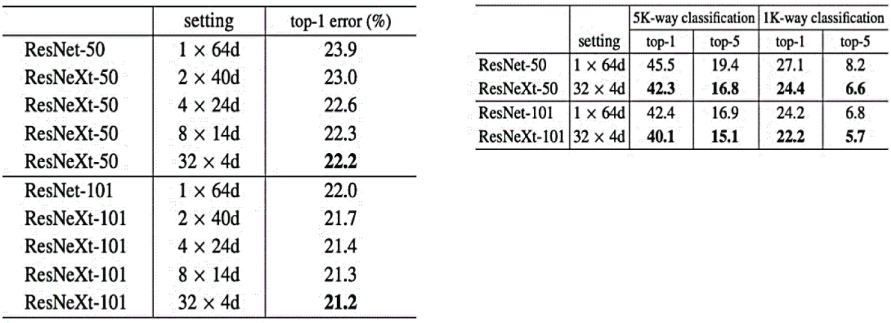


Figure 4. A comparative view between the parameters of ResNet and those of ResNeXt (Xie et al., 2017)



stage	output	ResNet-50	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**×10^6	**25.0**×10^6
FLOPs		**4.1**×10^9	**4.2**×10^9

Figure 5. Different parameters of 2 × 2 complexity models



	setting	top-1 error (%)
ResNet-50	1 × 64d	23.9
ResNeXt-50	2 × 40d	23.0
ResNeXt-50	4 × 24d	22.6
ResNeXt-50	8 × 14d	22.3
ResNeXt-50	32 × 4d	**22.2**
ResNet-101	1 × 64d	22.0
ResNeXt-101	2 × 40d	21.7
ResNeXt-101	4 × 24d	21.4
ResNeXt-101	8 × 14d	21.3
ResNeXt-101	32 × 4d	**21.2**

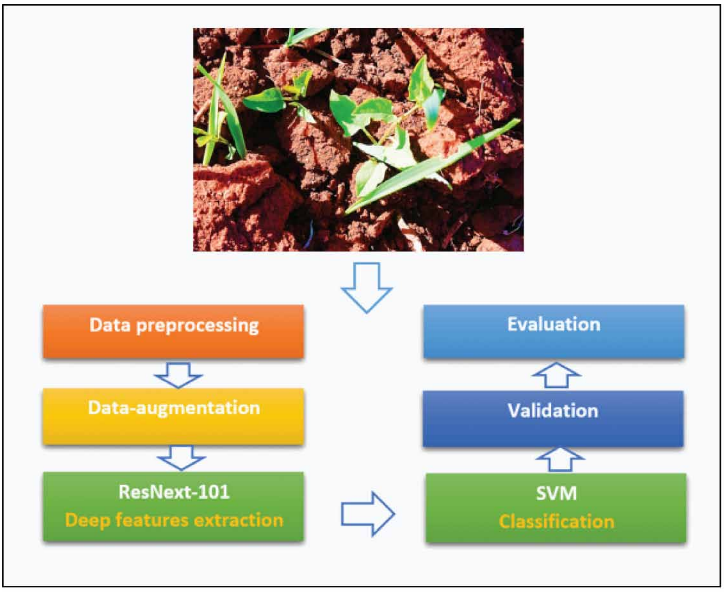
		5K-way classification		1K-way classification	
	setting	top-1	top-5	top-1	top-5
ResNet-50	1 × 64d	45.5	19.4	27.1	8.2
ResNeXt-50	32 × 4d	**42.3**	**16.8**	**24.4**	**6.6**
ResNet-101	1 × 64d	42.4	16.9	24.2	6.8
ResNeXt-101	32 × 4d	**40.1**	**15.1**	**22.2**	**5.7**

is high. In addition, compared to ResNet, ResNeXt performs well in terms of precision. This is why we chose ResNeXt50 in a science experiment, the details of which we will discuss in the next part.

Weed Detection Using a New Hybrid Model

In agricultural applications, accurate prediction results are crucial, as any prediction error can lead to huge losses and wasted herbicide. Thus, to further increase the precision of the results and to make strong predictions, we have implemented training by the method of incremental pattern recognition. The basic model applied here, as shown in Figure 6, is based on ResNeXt-50 pre-trained on our dataset. The pre-trained model is then refined for dataset and used for the reason of extracting features necessary in depth images belong to four classes. Thus, the deep learning architecture is applied to extract the characteristics of given images. The resulting features which are extracted are then classified by the first part of our model which is the SVM linear classifier which will predict whether an image is a weed or not and predict the type of this weed. This proposed network, according to the latest scientific publications, has not yet been addressed.

Figure 6. The proposed hybrid model



Regarding the parameters used, we can cite a batch size of four (Batch Size), the activation function is ReLU and Softmax, the optimizer is Adam. The model is trained on our dataset and the next part discusses the results obtained.

RESULTS AND DISCUSSION

Since the quantity of images containing in our dataset is average, we performed a technique which allows an increase of the data to obtain more images which makes obtaining interesting results. Accuracy was improved by the ResNeXt + SVM hybrid architecture, which has exceptionally good results with the accuracy of 98% for the detection of weeds, which is the problem of our study, as shown in Figures 7 and 8.

From the results presented in the graphs, we can conclude that the measurements adopted to measure the performance recorded on Tensorboard showed good accuracy reaching 98% (Figure

Figure 7. The accuracy of the model after training

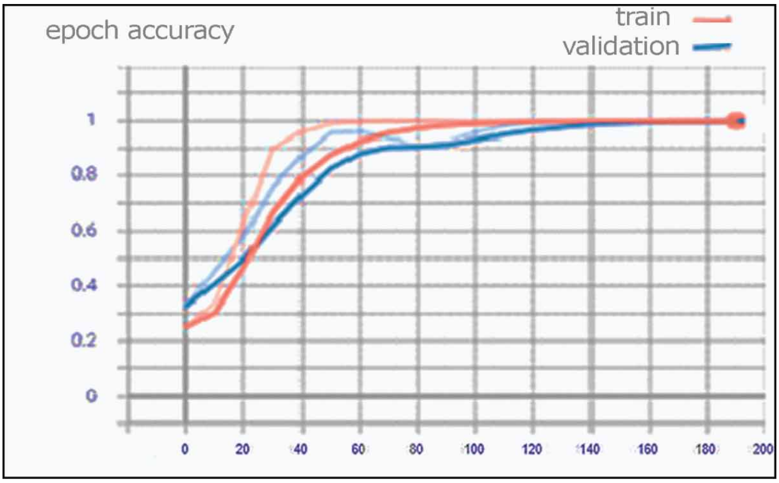
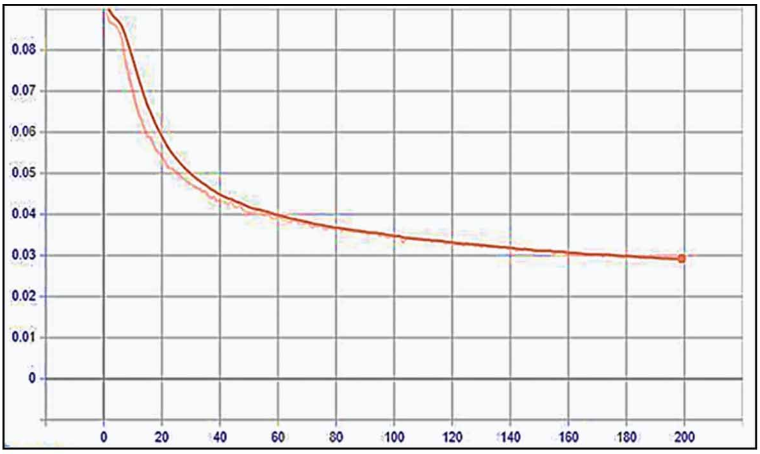


Figure 8. The loss of the model after learning



7). They also showed the decrease in the total error, which fell to 29% (figure 8). Learning this new ReXnet-SVM model is faster than the other models, around 4 hours of learning. This means that the performance of the model can still improve with each epoch if the epochs are increased. Another diagnostic tool is the presentation of the predictions on the test images, which gives in our case the same fact (Figure 9), the majority of these predictions, are well placed with a rate of 98%, and you find above a model of these predictions.

The model gave a good training accuracy of 98% and a decent validation accuracy of 97%, which is definitely better than the other models subject of comparison; ResNeXt, ResNet and VGG16 (Dos Santos, 2017; Asad, 2019; Kim, 2021), we compared them with the proposed method in terms of precision. The training loss gradually decreased from an initial value of 0.9 to its optimum and 0.28 as the lower value. With the application of incremental learning thanks to the first part of our model, presented by the ResNeXt architecture, which is used for the in-depth feature extraction and the second part consisting of SVM linear classifier, which is used for the task classification. The results are interesting in terms of precision compared to other methods used in the same problem our study. It also gave better validation accuracy and better training accuracy. This ResNeXt + SVM model also gave good predictions, which are accurate regarding weed detection on the test images (Figure 9). We compared our approach with the models we have discussed in related work. Table 1 shows the precision of each model (Dos Santos, 2017; Kim, 2021; Lammie, 2019) as well as the result of our model which gave an important precision compared to the results of the comparison models ResNet and ResNeXt, our model gave important results compared with the results of the models under comparison.

The proposed deep learning approach for weed classification has major contributions over the methods cited in the literature, as follows.

First contribution: To our knowledge, this is the first application of the combined model of ResNeXt and SVM in weed classification. This combination model can effectively use the temporal and spatial characteristics of information to achieve better classification results.

Figure 9. Predictions on a test image

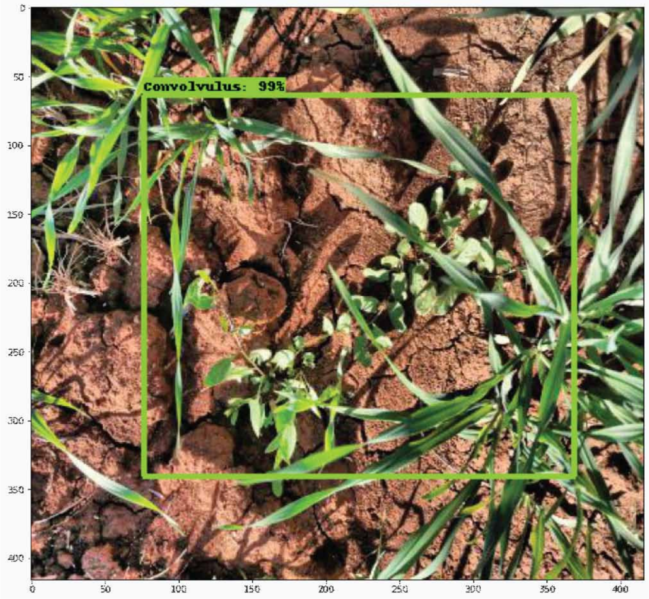


Table 1. Comparison of the proposed model with ResNet and ResneXt

Description of the research problem	Data	Model	Precision
Detection and classification of weeds in the crop	Crop images captured with a drone	ResNet	95,1%
		ResNeXt	97,4%
	Crop Images captured with professional camera. Crop images in public datasets.	The proposed Model	98%

Second contribution: The use of the pre-trained method of the second part ResNeXt, allows the choice of its weights, which are close to a better local optimum, it allows the weights to be maintained in a higher gradient range and can be refined perfectly.

The last contribution: the ResNeXt + SVM model is more efficient; more affordable; easy to work with about 98% accuracy this would result in effective weed detection. Therefore, we can use it in a system of weed recognition that helps remove weeds in an automated manner that saves pesticides and protects the environment, since the spray is topical (Jabir, et al., 2021).

CONCLUSION

When we talk about precision agriculture, it encompasses a range of fields, and we have limited our study to techniques for detecting and identifying plants and weeds. The adoption and development of precision agriculture require new models of deep learning. In this paper, we have proposed a hybrid classifier which is made by a balanced combination of machine learning and deep learning and which has been implemented for feature extraction and classification for weed detection in agricultural fields. Our model suggested as a new solution is very precise compared to other standard deep learning architectures known in the field of bad Hebrews, in terms of accuracy and several other evaluation metrics. The ResNXt-SVM hybrid model could provide accurate predictions, its accuracy is up to 98% that is better than ResNet and ResNext whose accuracy results are 95.1% and 97.4%. Thus, this model can be applied to any type and size of dataset related to different domains that require very precise decisions. However, the accuracy of the model can be further improved by applying machine learning class-balancing methods.

CONTRIBUTION

Authors are equally related to the writing of the manuscript and are equally responsible for plagiarism.

CONFLICT OF INTEREST

The authors declare no conflict of interests regarding the publication of this article.

ACKNOWLEDGMENT

This research is part of a scientific project to prepare a doctorate, in which a group of doctors from the LIMATI Laboratory at Sultan Moulay Slimane University in Morocco are involved. The general topic of this project is: Deep learning for the benefit of the digitization of agriculture.

REFERENCES

Agarap, A. F. M. (2018, February). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In *Proceedings of the 2018 10th international conference on machine learning and computing* (pp. 26-30). doi:10.1145/3195106.3195117

Alalshekmubarak, A., & Smith, L. S. (2013, March). A novel approach combining recurrent neural network and support vector machines for time series classification. In *2013 9th International Conference on Innovations in Information Technology (IIT)* (pp. 42-47). IEEE. doi:10.1109/Innovations.2013.6544391

Asad, M. H., & Bais, A. (2019). Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. *Information Processing in Agriculture*.

Bansal, N., Chen, X., & Wang, Z. (2018). *Can we gain more from orthogonality regularizations in training deep CNNs?* arXiv preprint arXiv:1810.09102.

Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, *44*(1), 108–132. doi:10.1006/jmps.1999.1279 PMID:10733860

Choudhary, R., & Gianey, H. K. (2017, December). Comprehensive review on supervised machine learning algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)* (pp. 37-43). IEEE. doi:10.1109/MLDS.2017.11

Chuck, C., Laskey, M., Krishnan, S., Joshi, R., Fox, R., & Goldberg, K. (2017, August). Statistical data cleaning for deep learning of automation tasks from demonstrations. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)* (pp. 1142-1149). IEEE doi:10.1109/COASE.2017.8256258

Deng, L., & Liu, Y. (2018). A joint introduction to natural language processing and to deep learning. In *Deep learning in natural language processing* (pp. 1–22). Springer. doi:10.1007/978-981-10-5209-5_1

Deng, L., Yu, D., & Platt, J. (2012, March). Scalable stacking and learning for building deep architectures. In *2012 IEEE International conference on Acoustics, speech and signal processing (ICASSP)* (pp. 2133-2136). IEEE. doi:10.1109/ICASSP.2012.6288333

Doreswamy, D., & Nigus, M. (2020, March). Feature Selection Methods for Household Food Insecurity Classification. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)* (pp. 1-7). IEEE. doi:10.1109/ICCSEA49143.2020.9132945

Dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., & Folhes, M. T. (2017). Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture*, *143*, 314–324. doi:10.1016/j.compag.2017.10.027

Forero, M. G., Herrera-Rivera, S., Ávila-Navarro, J., Franco, C. A., Rasmussen, J., & Nielsen, J. (2018, November). Color classification methods for perennial weed detection in cereal crops. In *Iberoamerican Congress on Pattern Recognition* (pp. 117–123). Springer.

Hao, J., & Ho, T. K. (2019). Machine learning made easy: A review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, *44*(3), 348–361. doi:10.3102/1076998619832248

Hitawala, S. (2018). *Evaluating ResNeXt Model Architecture for Image Classification*. arXiv preprint arXiv:1805.08700.

Hoiem, D., Gupta, T., Li, Z., & Shlapentokh-Rothman, M. (2021, July). Learning Curves for Analysis of Deep Networks. In *International Conference on Machine Learning* (pp. 4287-4296). PMLR.

Hothorn, T., & Lausen, B. (2003). Double-bagging: Combining classifiers by bootstrap aggregation. *Pattern Recognition*, *36*(6), 1303–1309. doi:10.1016/S0031-3203(02)00169-3

Huang, L., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2019). Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4874-4883). doi:10.1109/CVPR.2019.00501

Jabir, B., Falih, N., & Rahmani, K. (2021). Accuracy and Efficiency Comparison of Object Detection Open-Source Models. International Journal of Online & Biomedical Engineering, 17(5), 165. doi:10.3991/ijoe.v17i05.21833 doi:10.3991/ijoe.v17i05.21833

Jabir, B., Falih, N., & Rahmani, K. (2021). Deep learning-based decision support system for weeds detection in wheat fields. Iranian Journal of Electrical and Computer Engineering, 12(1), 979–990. 10.11591/ijece.v12i1.pp%25p

Jabir, B., Falih, N., Sarih, A., & Tannouche, A. (2021). A Strategic Analytics Using Convolutional Neural Networks for Weed Identification in Sugar Beet Fields. AGRIS On-Line Papers in Economics and Informatics, 13(1), 49-57. doi:10.7160/aol.2021.13010410.7160/aol.2021.130104

Jahrer, M., Töscher, A., & Legenstein, R. (2010, July). Combining predictions for accurate recommender systems. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 693-702). doi:10.1145/1835804.1835893 doi:10.1145/1835804.1835893

Johnson, J. M., & Khoshgoftaar, T. M. (2020). The effects of data sampling with deep learning and highly imbalanced big data. Information Systems Frontiers, 22(5), 1113–1131. doi:10.1007/s10796-020-10022-7 doi:10.1007/s10796-020-10022-7

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. doi:10.1016/j.compag.2018.02.016

Kao, Y. T., Zahara, E., & Kao, I. W. (2008). A hybridized approach to data clustering. *Expert Systems with Applications*, *34*(3), 1754–1762. doi:10.1016/j.eswa.2007.01.028

Kim, M. S., Lee, G. Y., & Kim, H. G. (2021). Exotic Weed Image Recognition System Based on ResNeXt Model. *Journal of Korea Multimedia Society*, *24*(6), 745–752.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering, 160*(1), 3-24.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159–190. doi:10.1007/s10462-007-9052-3

Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017, December). Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)* (pp. 364-371). IEEE.

Lammie, C., Olsen, A., Carrick, T., & Azghadi, M. R. (2019). Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 51171–51184. doi:10.1109/ACCESS.2019.2911709

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., & Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(9), 6690–6709. doi:10.1109/TGRS.2019.2907932

Li, T., Levina, E., & Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika*, *107*(2), 257–276. doi:10.1093/biomet/asaa006

Mahajan, A., & Chaudhary, S. (2019, June). Categorical image classification based on representational deep network (RESNET). In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 327-330). IEEE.

Mi, X., Zou, F., & Zhu, R. (2019). Bagging and deep learning in optimal individualized treatment rules. *Biometrics*, *75*(2), 674–684. doi:10.1111/biom.12990 PMID:30365175

Milioto, A., Lottes, P., & Stachniss, C. (2018, May). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 2229-2235). IEEE.

Muppala, C., & Guruviah, V. (2020). Machine vision detection of pests, diseases and weeds: A review. *Journal of Phytology*, *12*, 9–19. doi:10.25081/jp.2020.v12.6145

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., & Sutskever, I. (2019). *Deep double descent: Where bigger models and more data hurt.* arXiv preprint arXiv:1912.02292.

Narendran, D. J., Abilash, R., & Charulatha, B. S. (2021). Exploration of Classification Algorithms for Divorce Prediction. In *Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications* (pp. 291-303). Springer. doi:10.1007/978-981-15-7234-0_25

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 19143–19165. doi:10.1109/ACCESS.2019.2896880

O'Shea, A., Lightbody, G., Boylan, G., & Temko, A. (2018, July). Investigating the impact of CNN depth on neonatal seizure detection performance. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5862-5865). IEEE.

Olsen, A., Han, S., Calvert, B., Ridd, P., & Kenny, O. (2015, November). In situ leaf classification using histograms of oriented gradients. In *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (pp. 1-8). IEEE. doi:10.1109/DICTA.2015.7371274

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Azghadi, M. R., & White, R. D. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Scientific Reports*, *9*(1), 1–12. doi:10.1038/s41598-018-38343-3 PMID:30765729

Osband, I., Aslanides, J., & Cassirer, A. (2018). *Randomized prior functions for deep reinforcement learning.* arXiv preprint arXiv:1806.03335.

Ouyang, F. S., Guo, B. L., Ouyang, L. Z., Liu, Z. W., Lin, S. J., Meng, W., Huang, X., Chen, H., Qiu-gen, H., & Yang, S. M. (2019). Comparison between linear and nonlinear machine-learning algorithms for the classification of thyroid nodules. *European Journal of Radiology*, *113*, 251–257. doi:10.1016/j.ejrad.2019.02.029 PMID:30927956

Pal, K. K., & Sudeep, K. S. (2016, May). Preprocessing for image classification by convolutional neural networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 1778-1781). IEEE. doi:10.1109/RTEICT.2016.7808140

Pavlyshenko, B. M. (2016, August). Linear, machine learning and probabilistic approaches for time series analysis. In *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)* (pp. 377-381). IEEE. doi:10.1109/DSMP.2016.7583582

Pulido, C., Solaque, L., & Velasco, N. (2017). Weed recognition by SVM texture feature classification in outdoor vegetable crop images. *Ingenieria e Investigacion*, *37*(1), 68–74. doi:10.15446/ing.investig.v37n1.54703

Ranjan, R., Sankaranarayanan, S., Bansal, A., Bodla, N., Chen, J. C., Patel, V. M., Castillo, C. D., & Chellappa, R. (2018). Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, *35*(1), 66–83. doi:10.1109/MSP.2017.2764116

Reimers, N., & Gurevych, I. (2017). *Optimal hyperparameters for deep lstm-networks for sequence labeling tasks*. arXiv preprint arXiv:1707.06799.

Rice, L., Wong, E., & Kolter, Z. (2020, November). Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning* (pp. 8093-8104). PMLR.

Schreiber, J. (2017). Pomegranate: Fast and flexible probabilistic modeling in python. *Journal of Machine Learning Research*, *18*(1), 5992–5997.

Shi, H., Li, H., Zhang, D., Cheng, C., & Cao, X. (2018). An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification. *Computer Networks*, *132*, 81–98. doi:10.1016/j.comnet.2018.01.007

Steiger, J. H. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, *25*(2), 173–180. doi:10.1207/s15327906mbr2502_4 PMID:26794479

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (pp. 843-852). doi:10.1109/ICCV.2017.97

Thara, D. K., PremaSudha, B. G., & Xiong, F. (2019). Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. *Pattern Recognition Letters*, *128*, 544–550. doi:10.1016/j.patrec.2019.10.029

Van Tulder, G., & de Bruijne, M. (2015, October). Why does synthesized data improve multi-sequence classification? In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 531-538). Springer. doi:10.1007/978-3-319-24553-9_65

Vapnik, V., Guyon, I., & Hastie, T. (1995). Support vector machines. *Machine Learning*, *20*(3), 273–297. doi:10.1007/BF00994018

Vogelsang, D. C., & Erickson, B. J. (2020). *Magician's corner: 6*. TensorFlow and TensorBoard.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492-1500). doi:10.1109/CVPR.2017.634

Zhang, Z., Chen, B., Sun, J., & Luo, Y. (2021). A bagging dynamic deep learning network for diagnosing COVID-19. *Scientific Reports*, *11*(1), 1–15. doi:10.1038/s41598-021-95537-y PMID:34381079

Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., & Fang, W. (2020). Deep learning-based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics*, *43*, 101003. doi:10.1016/j.aei.2019.101003

Zhou, D., Tian, F., Tian, X., Sun, L., Huang, X., Zhao, F., Zhou, N., Chen, Z., Zhang, Q., Yang, M., Yang, Y., Guo, X., Li, Z., Liu, J., Wang, J., Wang, J., Wang, B., Zhang, G., Sun, B., & Li, X. et al. (2020). Diagnostic evaluation of a deep learning model for optical diagnosis of colorectal cancer. *Nature Communications*, *11*(1), 1–9. doi:10.1038/s41467-020-16777-6 PMID:32528084

Brahim Jabir received his doctorate degree in 2021 and his Master degree in 2015 in compute engineering and systems at the polydisciplinary Faculty of Sultan Moulay Slimane University in Beni Mellal, Morocco. Currently, he is a teacher in regional center for teaching professions in Beni Mellal, Morocco. His research interest is about Digital Agriculture, Deep learning, Strategic Analytics and Information Systems.

Noureddine Falih is PhD on Computer Science from Faculty of Sciences and Technologies of Mohammedia, Morocco in 2013. He is an associate professor in Polydisciplinary Faculty of Sultan Moulay Slimane University at Beni Mellal, Morocco since 2014. He has 18 years of professional experience in several renowned companies. His research topics are about Information System Governance, Business Intelligence, Big Data Analytics and Digital Agriculture.