International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Crops and weeds classification using Convolutional Neural Networks via optimization of transfer learning parameters

Abdel-Aziz Binguitcha-Fare, Prince Sharma

Abstract*: *Agriculture remains the backbone of several economies in the world, especially in underdeveloped countries. With the rapid growth of the population and the increasing demand in food, farmers need to maximize the productivity and one possibility is the reduction of losses. Weeds are one of the major dangers in farming.Indeed, they competevigorously with the crop for nutrients and water. As result, they can cause the loss of 10% to 100% of the total harvest. This work aimed at developing a new model tailored to classify crops and weeds images. Using a pubic dataset of 5339 plant images from Aarhus University Signal Processing group in collaboration with University of Southern Denmark, we proposed a methodology based on transfer learning technique to classify 12 species of crops and weeds. Firstly, we converted images to jpeg format in order to accelerate the convergence and data augmentation techniques such as resizing, rotating, flipping, scaling were employed to reduce the chances of overfitting. Then, a model trained on ImageNet dataset with Residual Network 101 architecture was used for performing transfer learning. Finally, the network's parameters were adjusted through various techniques involving progressive resizing, cyclical learning rate and focal loss function for improving the performance. Our model achieved an overall accuracy of 98,47% during validation and of 96,04% on the test set. We already deployed the model over Internet through a web application and our next step will be to integrate this solution within a mobile application and embedded devices.Future works concerns with the use of more features and descriptors to accurately distinguish two specificclasses of weeds: Black-grass and Loose Silky-Bent, and the possibility to extend our approach to other kinds of plants.

Index Terms*: *Precision Agriculture, Convolutional Neural Networks, Transfer Learning, Residual Networks.

I. INTRODUCTION

The rapid growth of the population and global warming are creating new challenges to the farmers. From one part, they need to increase the farming production and on theother hand, they must maximize profits. The world population is currently estimated at 7.7 billion and as per the report of FAO (Food and Agriculture Organization of the United Nations) [1], by 2050, the world's population will reach 9.1 billion. To satisfy this number of people, global food production will need to grow by 70%. Therefore, we need to make agriculture more intelligently forminimizing the losses. New technologies help in addressing these challenges. This revolution led to the emergence of a new term: Precision Agriculture or Precision Farming.

Precision Agriculture is the use of new technologies to improve the traditional agriculture by enabling farmers to analyze all of data associated to their farms. It involves the use of Global Positioning System (GPS), the sensors, the weather tracking and other technologies. By using all these materials, the farmers get powerful insights which will assist them in making decision. One of the main problems encountered in farming is the ability to recognize weeds among native species of plants. Typically, this task is performed by field personnel, either by walking or the use of a motor vehicle. Moreover, it is not evident to detect and recognize weeds between crops due to their strong similarities. However, it is worth noting that, successful cultivation of lot of plants is widely associated with the efficacy of weed control [2]. During the first six to eight weeks after seeding, weeds compete vigorously with the crop for nutrients and water. As result, annual yield losses arise in cultivated crops. The losses caused by weeds vary according to the type of weed, type of crop, and the environmental conditions involved. An accurate detection of weeds is of high importance to reduce losses and consequently increase the productivity. Major advances in Artificial Intelligence contributed widely to improve the weed detections. Deep Learning [3], a class of Machine Learning [4] algorithms achieved great results in complex problems. Compared to traditional machine learning models in which features are selected manually and extracted through well-defined algorithms, Deep Learning networks automatically, discover higher level features from data by themselves. This is possible through utilization of large amounts of data and high computational resources. Most of the time, it is difficult or even impossible to have the necessary amount of data or the computing power required. This led to the concept of transfer learning. As the name suggests, it consists to apply knowledge learned from a problem, involving massive amounts of data, to another problem, different but related to it. There are two ways toapply transfer learning in deep learning. The first approach consists to use a pre-trained network without its final layer as a fixed feature extractor for another task.

Manuscript published on 30 June 2019.
* Correspondence Author (s)
Abdel-Aziz Binguitcha-Fare, is a Research Scholar, AP Goyal Shimla University, School of Science and Technology, Shimla, HP, India.
Prince Sharma, is an Assistant Professor, AP Goyal Shimla University, School of Science and Technology, Shimla, HP, India.

The outputs of the layer prior the last fully connected layer, constitute the features extracted.A second approach consists to fine-tune the network weights by freezing and unfreezing certain or the entire layers of the model. In this paper, we presented a deep learning-based approach system to classify nine (09) species of crops: Black-grass, Charlock, Cleavers, Common Chickweed, Fat Hen, Loose Silky-bent, Scentless Mayweed,
Shepherd's Purse, Small-flowered Cranesbill and three (03) species of weeds namely: Common wheat, Maize, Sugar beet. Our systemwas powered by a pre-trained deep learning model: Residual Network 101 (ResNet 101).The main contribution of our work was to study and develop an efficient solution for automatically classify crops and weeds at several growth stages by investigating variants novel techniques such as progressive resizing, cyclical learning rates and focal loss. The combination of all these methods served to achieve an overallaccuracy of 98.47% on the validation set and roughly96.04% on the test set. Our model was deployed as web application available at the following link: http://weeds-detection.abtech.ovhand can be used by farmers to recognize weeds and crops. Throughout our experiments, we did also some observations that could inspire other researchers when they would perform a similar work. The rest of this paper was organized as follows: The next section presents an overview of related work and the section 3 will present our proposed methodology. The section 4 is focused on the experiments as well as the results. Finally, conclusion was offered in the last section.

II. LITERATURE REVIEW

In literature, various approaches and systems have been proposed to classify crops and weeds.In [5], the authors tried to address this problem by using histogram based on color indices to discriminate between three classes: soil, soybean and broadleaf (weeds). The feature representation was tested with two powerful classifiers namely Back-propagation Neural Network (BPNN) and Support Vector Machine (SVM). This approach achieved an overall accuracy of 96% and 95% respectively. Novel models such as VGGNet, DetectNet, GoogLeNet achieved high accuracy at detecting weeds in Bermuda grass turfgrasses. exhibited high f1 score values over 95% [6]. Another study discussed in [7] proposed a fully Convolutional Neural Network (CNN) with encoder-decoder structure and incorporates spatial information by considering image sequences. In details, this solution encodes the spatial arrangement of plants in a row using 3D convolutions over an image sequence. The experiments demonstrated that the system generalizes better to unseen fields in comparison to other state-of-the-art approaches. In another paper [8], the authors studied the use of CNNs with unsupervised training dataset collection for weeds detection fromimages captured with drones. An Area-Under-Curve(AUC) over 80% was achieved while performing tests with bean and spinach. The study presented in [9] focused on the broad leaf weed detection in pasture. CNN and a Quadric-SVM (Support Vector Machine) models were considered for conducting analyses and showed incredible accuracies of 96.88% and 89.4% respectively. A use of SVM with 18 feature descriptors, permitted to classify 11 plant species including plants and weeds with a precision of 93% in [10]. In the study [11], the authors analyzed the performance of a fully trained network for classifying plant seedling images into twelve species (weeds and plants). The model discussed, achieved an excellent performance with an accuracy rate of 90.15%. The paper [12] proposed a CNN-based method for estimating the growth stage in terms of number of leaves of various weed species. An average accuracy for these species was 70%. For detecting broadleaf and grass weeds in relation to soil and soybean, the authors of [13] investigated also the use of CNNs. Their works achieved above 98% accuracy with an accuracy average between all images above 99%. Traditional machine learning algorithms and deep learning models were compared for seedling classification in the paper [14]. A good accuracy of 92.6% has been obtained by performing background segmentation.
The authors of [15] demonstrated the great performance of CNNs to learn useful features representations for 44 different plant species with high precision. Currently, there have been numerous researches aiming todevelop automated analysis of plant images. Complete reviews of various techniques used [16, 17] have been published last year. Since 2012 with the emergence of AlexNet, CNNs models became widespread and capable of addressing image classification problem in multiple domains [18, 19]. In the papers studied, we noticed that, researchers trained the models from scratch and rarely perform optimization of hyperparameters. In addition, the works conducted on the same dataset did not exceed an accuracy of 94%. Thus, the purpose of this work is to investigate the residual networks and several optimization functions in order to provide a better model.

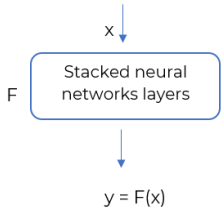
III. PROPOSED METHODOLOGY

Our solution was based on the CNN architecture ResNet101 that was pre-trained on ImageNet database. In this section, we will give a complete overview of our approach.

III.1 ResNet Overview

ResNet, short name for Residual networks is arguably one of the most groundbreaking work in computer vision tasks. This model presented in [20] by some researchers at Microsoft Research, allow us to train extremely deep neural networks with hundred layers successfully. Prior to this novel architecture, training very deep neural networks suffered from vanishing gradients problems [21]. Indeed, increasing network depth does not work by simply pushing layers together. As the gradients are back-propagated to earlier layers, the repeated operations may make the gradients smaller and as result, the overall performance of the network can progressively decrease. The ResNet model addresses this problem by introducing a new concept called Skip Connection. The following figure (figure1)shows a traditional block of convolution layers whereas the second one (figure 2) depicts the skip connection technique.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)



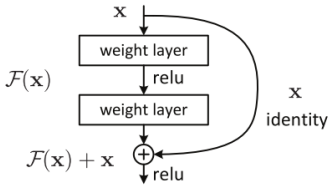


Fig.1: Traditional block **Fig.2: Skip Connection**

The core idea of this novel architecture is to add the original input to the output of the convolution stack. This solution works because it allows to flow information from earlier layers in the model to later layers by using an alternate path if necessary. Multiple versions exist according to the number of layers. We can state Resnet18, ResNet34, ResNet50, ResNet101, ResNet152. In our approach, after multiple analyses and experiments, we decided to work with ResNet101.

III.2 Network architecture

The architecture of ResNetwas described on the figure 3.

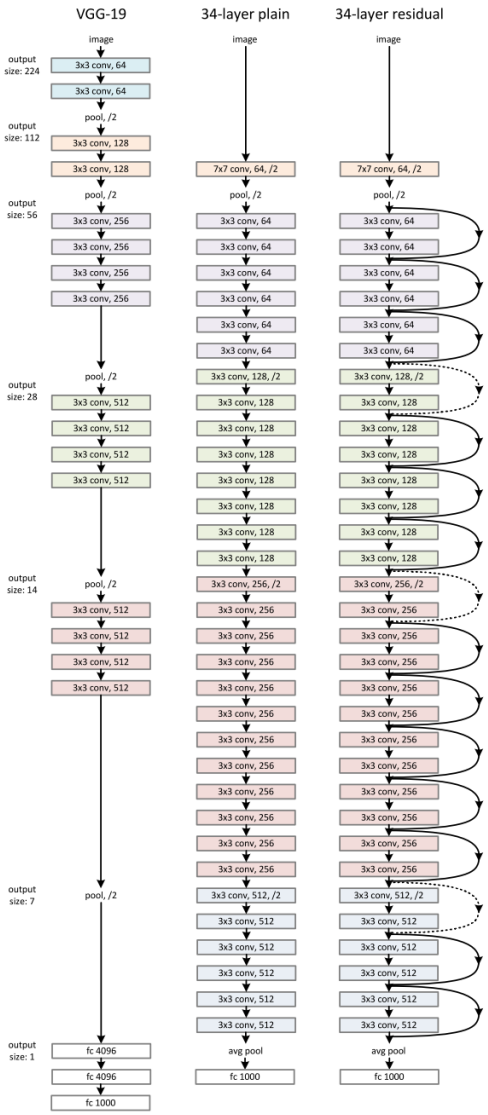


Fig. 3: Residual Networks Structure [20]

Typically, the model consists of blocks (convolution and identity), average pool layer and a fully connected layer with SoftMax as activation function.

III.2.1 Convolutional layers

They are definitely the most important blocks of CNNs. Their main objective consists to extract features from input images by convolving a set of features to align with the input feature map.

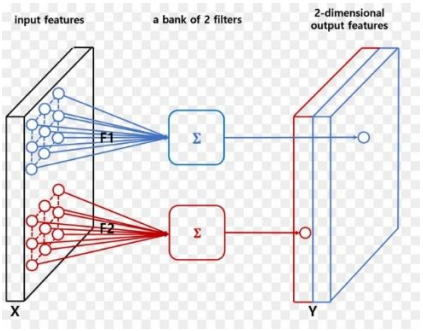


Fig. 4: Convolution process [22]

As one can see, a convolution layer comprises a set of independent filters (2 in our case). Each one is independently convolved with the image and produces a set of feature maps.

III.2.2 Activation function

When dealing with neural networks, a common practice consists to apply a nonlinear activation function at a node of the network. This produces an output which will be used by the next node and so on until a desired solution is found. In the case of CNN, we introduce an activation function after each of the convolutional layers. One of the most used in computer vision is the Rectified Linear Unit (ReLU) [23]. It is widely popular due to its fat computation time, and because it prevents vanishing gradient problems.

III.2.3 Average pooling layer

In ResNet101 architecture, an average pooling layer is placed after the 5 blocks of convolution to perform a down-sampling by dividing the input into square region and then computing the average values of each region. Its main purpose is to reduce the dimensionality.

III.2.4 Fully connected layer

This part is in principle the same as a regular neural network. The features previously extracted are used to feed the network. At this point, there are already enough information for a fair degree of accuracy in recognizing classes. The final SoftMax layer applies a normalized exponential function to a K-dimensional vector, for K >=2, where each value of the vector is scaled in the range of (0, 1) and the sum of all values is equal to 1. In other words, it determines the confidence in the predicted class for each image.

III.3 Transfer learning

As we know, humans have an inherent ability to transfer knowledge across tasks. The knowledge acquired while learning about one task can be used to solve related tasks. The more related the tasks, the easier it will be for us to transfer our knowledge. In the same logic, we can leverage knowledge such as features, weights from previously trained models for training newer models and even tackle problems like having less data for the newer task [24].

Indeed, the algorithms of deep learning needs a lot of data and long time to training the different weights and the million parameters of network (1.7 Million parameters in the case of ResNet 101). A such network requires large datasets and Graphical Processing Unit (GPU) for training process. In practice, it's difficult and expensive to join these two factors. One of the solutions that reduces effort is to take advantage of transfer learning, which provides guaranteed solution for an accurate classification with less training samples [25].
The transfer learning is based on the concept that, instead of training a deep neural network from scratch for a given task, we can take a network trained on a different domain for a different source tasks and then, adapt it for our domain and our target task (figure 4).

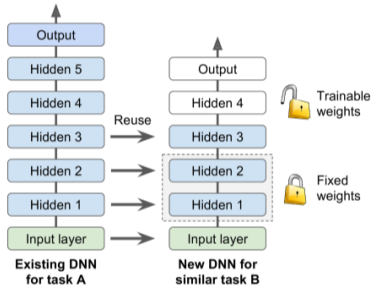


Fig 5: Transfer learning explained [25]

Depending on the context and the target dataset size, there are various transfer learning strategies. The two most popular strategies are: Off-the-shelf features and Fine Tuning [24].

- **Off-the-shelf features:** Deep Learning systems and models are powered by architectures that learn different features at different layers (hierarchical representations of layered features). These layers are finally connected to a last layer to get the final output. This layered architecture has an interesting advantage: It allows to utilize a pre-trained network (such as our ResNet101) without its final layer as a fixed feature extractor for other tasks. The main idea is to just leverage the pre-trained model's weighted layers to extract features but not to update the weights of the model's layers during training with new data for the new task.

- **Fine Tuning:** It is more relevant when the target dataset is very large. In this case, we do not just replace the finallayer, but we also selectively retrain some of the previous layers. Following this idea, we may freeze (fix weights) certain layers while retraining, or fine tune the rest of them to suit our needs. The training will be faster compared to a training from scratch because the parameters of all the layers (except the final one) are initialized with those of the pre-trained network.

III.4 System overview

In the following lines, we will discuss the fundamentals components of our approach.

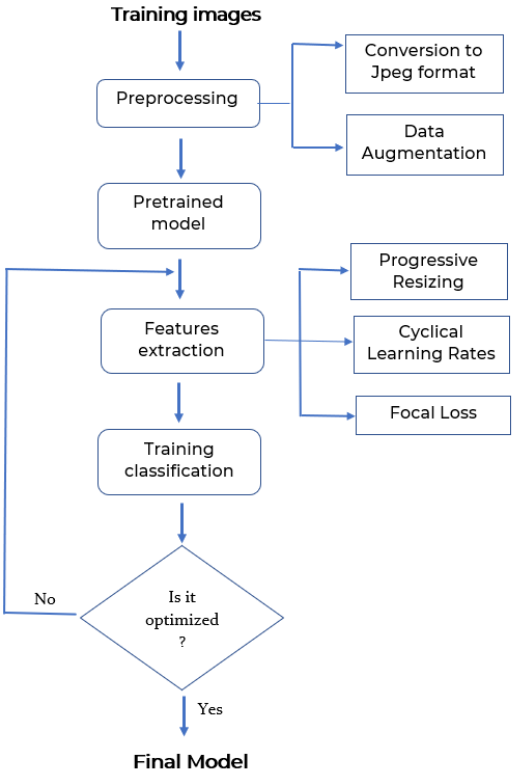


Fig. 6: Flow Diagram of proposed approach

III.4.1 Preprocessing

Our first task consisted to ensure that the input images were in Jpeg format. Indeed, the image quality can affect deep neural networks [26]. Many studies tend to prove that Jpeg [27] images provide a faster and more accurate CNN compared to other formats especially in the case of Residual Networks architectures [28]. The following task was data augmentation. Most of the time, the researchers develop their own dataset and the data are rarely available online. In our case, we have to deal with an imbalanced dataset. In order to reduce overfitting, data augmentation can be applied to artificially generate additional training data via transformations of existing training examples. The most common ones are affine transformations (horizontal and/or vertical flip, rotation). There are also non-affine transformations such as resizing, random crop (random part of an image), brightness and contrast variation, wrap (perspective), jitter (random noise) and cut-out (random black squares).

III.4.2 Pre-trained model

A pre-trained Model is a saved model that was already trained on large datasets such as ImageNet. This latter is a massive dataset with over 1 million labelled images pertaining to 1000 categories [29]. Once trained, these models work astonishingly well as feature detectors for mages they weren't trained on. Fortunately, many of the state-of-the art deep learning architectures have been openly shared by their respective teams. The pre-trained models are usually shared in the form of millions of parameters/weights and are available for everyone to use through different ways.

In our case, a pre-trained ResNet101 model is already available at the link [30] and has been used for our task.

III.4.3 Feature extraction

The process is illustrated by the figure 7.

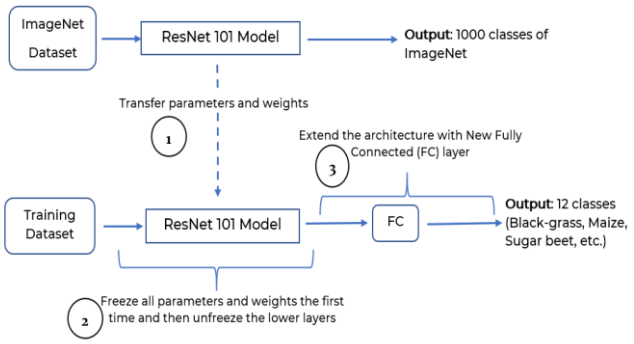


Fig.7: Feature extractor

The ResNet architecture trained with ImageNet was used. However, it is worth noting that, ImageNet dataset contains generic images of plants. Thus, a pre-trained model would not perform well for crops and weeds classification. Consequently, an adaptation was necessary to achieve the desired result. We proposed a mixture of the two strategies discussed early (Fine Tuning and Off-the-shelf features) to offer a new adaptation. Firstly, we started by keeping frozen all the layers of the pre-trained model so the weights already available were used for feature extraction and furthermore for the classification task. Once the last layers were producing good results, we unfrozen the lower layers and led various tests to determine the parameters which produce the best performance. In order to determine the optimum learning rate and achieve the best accuracy, we investigated three (03) techniques namely Progressive Resizing, Cyclical Learning Rates, and Focal Loss.

III.4.3.1 Progressive resizing

It's a technique to sequentially resize all the images while training from smaller to bigger sizes. As you know, CNNs build gradually higher-level features out of groups of pixels. The first layers capture parts, edges, so on and throughout the learning, the final layers predict the image class. However, a common problem encountered when building a CNN is the size of input images. Most of the time, images of dataset have different dimensions. The choice of input size must be carefully chosen because it has a direct impact on the time of convergence and the overall performance of the network. CNN can only work with standardly sized images; however too small images must be scaled up and too large ones must be scaled down. Some experiments [31] proved that, when a model is trained on very small images, the features learned are most important compared to those captured on very large images. Moreover, small images models are much faster to train and a model that performs correctly on small images will generalize to larger ones. The intuition behind the progressive resizing is to train the model with smaller input sizes, that will take less time to train and then, scaling up the images gradually until the desired result. A great way to use this technique is to train a model with smaller image size say 64x64, then use the weights of this model to train another model on images of size 96x96 and so on. Each larger-scale incorporates the previous smaller-scale model layers and weights in its architecture.

For our model, we started by building a simple classifier that performs quite well on 128x128 pixels. Then we scaled our model up to 196x196, 224x224, 299x299, 336x336, 350x350 and finally 360x360px. We noticed an important evolution in network performance during this process.

III.4.3.2 Cyclical learning rates

When training deep neural networks, the learning rate is probably the most important hyper-parameter to tune. It establishes the magnitude of weights updates. Its value can accelerate or slow down the convergence of the network. An optimum value appears crucial in order to improve classification accuracy. There is no fixed learning rate for a neural network, but an optimum learning rate is which one, leads to considerable decreases in the loss function. Traditionally, the common method consists to decrease progressively the value of learning until it becomes satisfactory for our needs. However, this method is not efficient and requires much time and effort. The paper [32] presents a novel method for defining the learning rate, named cyclical learning rates, which practically does without the need to perform several experiments to find the best values with no additional computation. Indeed, this technique lets the learning rate cyclically vary within a range of values instead of setting it to a defined value. It generally, allows to train without a need to tune and often in fewer iterations. This idea is presented through a triangular form (linearly increasing then linearly decreasing) because of its simplicity. This policy is also called triangular learning rate policy. Its generalized implementation is:

$$local\ cycle = math.floor(1 + epochCounter / (2 * stepsize))$$
$$local\ x = math.abs(epochCounter / stepsize - 2 * cycle + 1)$$
$$local\ lr = opt.LR + (\max LR - opt.LR) * math.\max(0, (1 - x))$$

Fig. 8: Cyclical learning rate function [32]

Where:

- opt.LR is the specified lower learning rate,
- epochCounter is the number of epochs of training.
- lr is the computer learning rate,
- stepsize is half the period or cycle length and
- maxLR is the maximum learning rate boundary.

Moreover, this method simplifies the decision of when to drop learning rates or to stop the current training. The author proposed the following convenient way to decide the learning rate range:

- Run the model for several epochs while letting the learning rate increase linearly (use triangular learning rate policy) between the lowest and the highest learning rate values.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

- Plot the accuracy versus learning rate curve. Note the learning rate value when the accuracy starts to increase and when the accuracy slows or starts to fall. These two learning rates are good choices for defining the range of the learning rates.

III.4.3.3 Focal loss

In most of machine learning tasks, we need to measure how bad our model is. This is done with a specific function called "loss function". During the training phase, our main objective consists to minimize this function and consequently increase our number of correct predictions. Calculating the losses of classification is possible with cross-entropy. This latter measure how well a set of estimated class probabilities match the targeted classes and is expressed with the following formula (figure 9).

$$H_p(q) = -\sum_{c=1}^{C} q(y_c) \cdot log(p(y_c))$$

Fig. 9: Cross-entropy formula [33]

Focal loss [34] is a novel loss function based on cross-entropy originally developed by Facebook AI Research for RetinaNet (one-stage object detector). One of the major differences between one-stage detector and two-stages detector is that, the first one uses a fixed grid of boxes (anchors) while the second one uses proposal network to generate (or to filter) box proposals. So, for the same image, more box proposals mean more background boxes. Normally, background boxes are easier to classify. Too many of them will bias the classifier to emphasize the background in order to minimize the overall box. The objects in the image will be down weighted. Focal Loss was proposed to improve the precision in this kind of tasks.In addition, it can be used in classification tasks when the data are imbalanced. Neural networks tend to be excessively confident in themselves. Focal Loss address this flaw by focusing the attention of neural network on the wrong predictions. Instead of trying to reduce predictions where the model's predictions are too far from the right values, it reduces the weight of the values it predicted correctly wrong. In other words, Focal Loss lowers the loss for well classified cases, while emphasizing hard ones by adding a factor to the standard cross entropy criterion. In practice, it is expressed through the following formula (Figure 10).

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t).$$

Fig. 10: Focal loss function [34]

III.4.4 Performance Evaluation

Accuracy is the most common metric used to evaluate the performance of a model. It corresponds to the total number of correct predictions among the total set of data. In a case of classification with multiple classes, we need to distinguish the average accuracy of overall accuracy. The first one is the mean of all the accuracies per class but the second one is the measure of accuracy over the whole set of data. Consequently, the overall accuracy is more realistic compared to the average accuracy and has been preferred throughout this work. Moreover, the accuracy metric makes really sense when the classes are uniformly distributed. In our case of multi-class classification with imbalanced data, it appears incorrect to rely on only this metric for measuring the efficiency of our model. So, we used other metrics in addition to accuracy in order to have the right performance of our solution. Firstly, we considered the following terms to describe the various proportions of our predictions: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). The TP concerns with the actual predicted plant class category that was successively classified. The FP pertains to other types of plant wrongly classified as the actual plant type. The TN relates to other classes that do not belong to the actual class. Finally, the FN pertains to the actual class category that was wrongly classified and did not belong to the actual class. The overall accuracy was computed using the equation (figure 11). Three metrics have been also used. The precision (figure 12) corresponds to the proportion of plants correctly identified as positive out of total plants of the plant class whereas the recall (figure 13) determines the number of plants correctly identified as positive out of total true positives items. The final one, f1-score (figure 14) is a kind of harmonic mean of precision and recall. It gives a balance between Precision and Recall.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \qquad Precision = \frac{TP}{TP+FP}$$

Fig.11: Overall accuracy **Fig. 12: Precision**

$$Recall = \frac{TP}{TP+FN} \qquad F1 = \frac{2*(Precision*Recall)}{(Precision+Recall)}$$

Fig. 13:Recall **Fig.14: F1 Score**

IV. EXPERIMENT RESULT AND DICUSSION

IV.1 Datasets

For conducting our experiments, we used a dataset containing approximately 5339 plant images constituted by Aarhus University Signal Processing group in collaboration with University of Southern Denmark [35]. This dataset presents 12 plant species at several growth stages. It comprises annotated RGB images with a physical resolution of roughly 10 pixels per mm. The following table (table I) shows the repartition of dataset. We can observe on the following figure (figure 15) an image of each specie. All the images of the dataset were in png format. We converted them into jpeg format to accelerate the training process. The dataset was divided into training set and test set with a ratio of 75%. After this operation, the new repartition of our dataset can be observed in the table II.

IV.2 Data augmentation

Data augmentation techniques have been performed for enlarge our training set and prevents overfitting. Indeed, the initial dataset was highly imbalanced.

Table I: Repartition of species

Specie	Number of elements	Type
Black-grass	309	Weed
Charlock	452	Weed
Cleavers	335	Weed
Common Chickweed	713	Weed
Common wheat	253	Crop
Fat Hen	538	Weed
Loose Silky-bent	762	Weed
Maize	257	Crop
Scentless Mayweed	607	Weed
Shepherd's Purse	274	Weed
Small-flowered Cranesbill	576	Weed
Sugar beet	463	Crop
Total	**5339**	

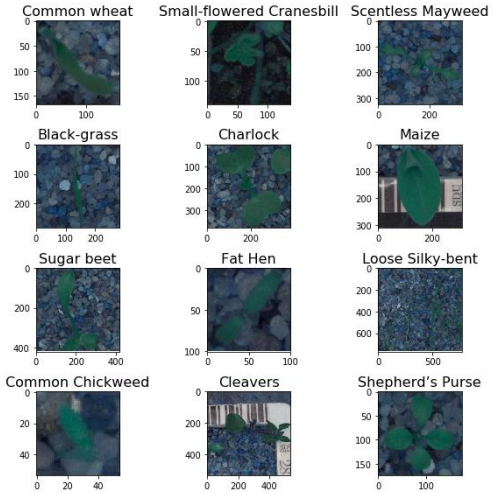


Fig.15: Images of each specie

Table II: Training and test sets

Specie	Training	Test
Black-grass	231	78
Charlock	339	113
Cleavers	251	84
Common Chickweed	534	179
Common wheat	189	64
Fat Hen	403	135
Loose Silky-bent	571	191
Maize	192	65
Scentless Mayweed	455	152
Shepherd's Purse	205	69
Small-flowered Cranesbill	432	144
Sugar beet	347	116
Total	**4149**	**1390**

By applying a variety of image transforms such as flip, rotate, ZCA whitening, Zoom. We increased all classes such that the exact number of images per class became 600. This number was chosen only for convenience. After all these operations, the training set was now 7200 images. It is worth noting that 20% of the training set has been considered as validation set. This permitted to measure the performance of the model during the training. So, the exact number of images dedicated for training was 5760 images and for validation was 1440.

The next table (table III) shows the final structure of our training data.

Table III: Training and validation sets

Specie	Initial training	Final data	
		Training	Validation
Black-grass	231	480	120
Charlock	339	480	120
Cleavers	251	480	120
Common Chickweed	534	480	120
Common wheat	189	480	120
Fat Hen	403	480	120
Loose Silky-bent	571	480	120
Maize	192	480	120
Scentless Mayweed	455	480	120
Shepherd's Purse	205	480	120
Small-flowered Cranesbill	432	480	120
Sugar beet	347	480	120
Total	**4149**	**5760**	**1440**

The following figure (figure 16) shows a sample of augmented plant images.

Figure 16: Sample of augmented plant images

IV.3 Training process

As we specified earlier, we used a residual network architecture ResNet 101 for performing our training. ResNet101 was been used to categorize over one million images (ImageNet database) into 1000 classes. Using transfer learning techniques, which is possible by reutilizing pre-trained model, we applied various optimizations for classifying plant images. Firstly, we started with an RGB color image of dimensions 128 x 128px in order to extract deep features and prevent scale dependency. We started with a learning of 1e-2 and throughout cyclical learning rate technique, we reduced progressively its value.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

In addition to this, we used focal loss technique to focus the attention of the network on the misclassified classes. The following table (table IV) presents a brief summary of all the parameters involved in the training process.

Table IV: Hyperparameters

Parameter	Value
Learning rate	1e-2
Cycle length	6
α	1.
ɣ	0.8

In addition to the learning, the network was trained with new dimensions of images while maintaining the weights obtained with the previous dimensions. By using this technique, we observed a better accuracy as the image get bigger.

IV.4 Results and evaluation

For our experiments, we used a powerful deep learning library called Fastai [36]. It is a free and open source project build on the top of PyTorch [37] for making cool neural networks. A single virtual machine with 2.30 GHz Intel(R) Xeon(R) CPU, 12 Gb RAM and NVIDIA Tesla K80 14 Gb was utilized via Google Colaboratory. The figures 17, 18 and 19 allow to observe the augmentation of accuracy when applying progressive resizing technique.

epoch	train_loss	valid_loss	accuracy	time
0	1.399428	0.774759	0.734722	01:29
1	0.748437	0.539503	0.821528	01:28
2	0.432554	0.371121	0.870833	01:29
3	0.265528	0.385226	0.880556	01:29
4	0.143532	0.386539	0.888889	01:29
5	0.084058	0.380902	0.890972	01:28

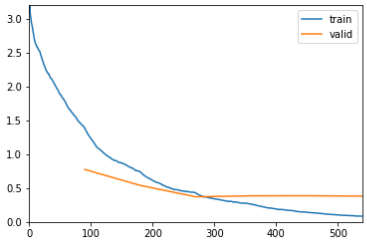


Fig. 17: Results obtained with 128x128 px

With the first one, we clearly observed an improvement from 73.47% to 89.09% on the validation test using an input image of 128px. Then, we noticed another jump from 76.65% to 93% with an image of 224px. Finally, we obtained an overall accuracy of 98.47% on an input image of 360 px. In addition to this, we can also notice the reduction of validation loss in each case of figure. That confirms the robustness of our solution and indicates that the model is not overfitting. The table (confusion matrix, figure 20) shows the confusion matrix obtained for the different classes during testing.

epoch	train_loss	valid_loss	accuracy	time
0	1.026296	0.954776	0.765556	01:56
1	0.626449	0.441194	0.838333	01:53
2	0.422232	0.299210	0.893333	01:54
3	0.280925	0.227598	0.913889	01:55
4	0.166465	0.216992	0.930000	01:53
5	0.102124	0.210090	0.929444	01:55

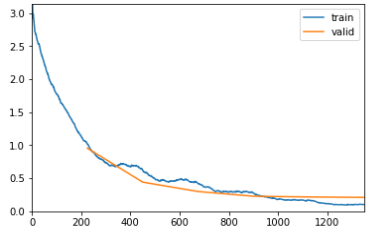


Fig. 18: Results obtained with 224x224 px

epoch	train_loss	valid_loss	accuracy	time
0	0.102948	0.048617	0.983333	07:43
1	0.115052	0.048393	0.984028	07:10
2	0.115437	0.048835	0.981250	07:09
3	0.112346	0.047835	0.984722	07:09
4	0.108952	0.048648	0.984028	07:09
5	0.099153	0.049194	0.984028	07:09

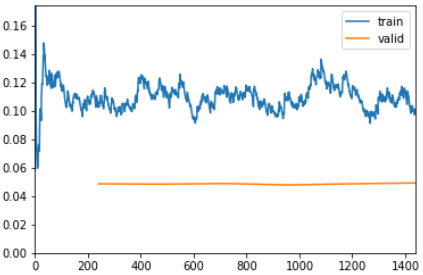


Fig. 19: Results obtained with 360x360 px

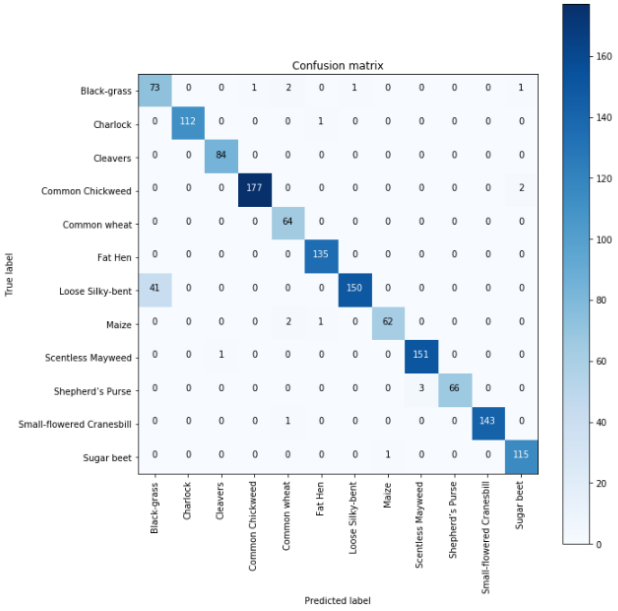


Fig.20: Confusion matrix for test set

It consists of twelve rows and columns that reports the number of false positives, false negatives, true positives, and true negatives.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

It explained the output for analysis of correct classifications with 1390 images. The rows specify the actual classification, and the columns correspond to the predicted rating. The table V summarizes all the results obtained for each category of plants.

model predicts a Black-grass and the second class predicted is Loose Silky-bent to find the average threshold above which the model can definitely choose the Black-grass or Loose silky-bent as the right predictionFor doing that, we proposed the following algorithm (fig. 22):

Table V: Results obtained for validation and testing

Specie	Validation				Test			
	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy
Black-grass	90.32%	95.75%	92.95%	98.81%	64.04%	93.59%	76.04%	96.66%
Charlock	100%	100%	100%	100%	100%	99.12%	99.56%	99.92%
Cleavers	100%	97.32%	98.64%	99.78%	98.82%	100%	99.41%	99.92%
Common Chickweed	98.35%	99.17%	98.76%	99.78%	99.44%	98.88%	99.16%	99.77%
Common wheat	99.22%	100%	99.61%	99.92%	92.75%	100%	96.24%	99.62%
Fat Hen	100%	99.22%	99.61%	99.92%	98.54	100%	99.26%	99.85%
Loose Silky-bent	95.80%	91.20%	93.44%	98.84%	99.34%	78.53%	87.72%	96.94%
Maize	100%	100%	100%	100%	98.41%	95.38%	96.88%	99.70%
Scentless Mayweed	99.10%	99.10%	99.10%	99.85%	98.05%	99.34%	98.69%	99.70%
Shepherd's Purse	99.07%	100%	99.53%	99.92%	100%	95.65%	97.78%	99.77%
Small-flowered Cranesbill	100%	100%	100%	100%	100%	99.31%	99.65%	99.92%
Sugar beet	100%	100%	100%	100%	97.46%	99.14%	99.29%	99.70%
Average values	**98.49%**	**98.48%**	**98.47%**	**99.74%**	**95.57%**	**96.58%**	**95.72%**	**99.70%**
Overall accuracies				**98.47%**				**96.04%**

It presents the classification results during final testing. After averaging, the final trained network yields an average accuracy of 99.70% and an overall accuracy of 96.04%. It is worth noting that, our model was able to classify the species except Black-grass and Loose Silky-bent with an accuracy of more than 99%. This means that all the crops species (Common wheat, Sugar beet, Maize) were correctly classified. In addition, we got precision of 100% for the species Charlock, Shepherd's Purse, Small-flowered Cranesbill and Sugar beet. We can notice also a high percentage of recall (100%) for Cleavers,Common wheat and Fat Hen. These results revealed a substantial performance and confirmed the efficiency of our model.

V. DISCUSSION

With the results obtained, we noticed a misclassification of the two plants: Black-grass and Loose Silk-bent. The first one is often classified as the second one. Although it is not a big problem because these two species are weeds, we decided to investigate and make some analyses in order to find the reasons of that.Firstly, we studied the top-2 predictions done by the model. We noticed that whenever, an image is correctly predicted as Black-grass, it was still with a high confidence (more than 99%) and 1% remaining is distributed across the 11 classes. But, when the prediction is a false positive, we noticed that the confidence is less or false and the second class with high rate of prediction was Loose Silky-bent. That supposes a high doubt of our model to clearly distinguish the first one class of the second one. So, the question is that: is there a threshold above which the model can affirm that, this specie is or is not Black-grass with more confidence?We conducted an experiment when the

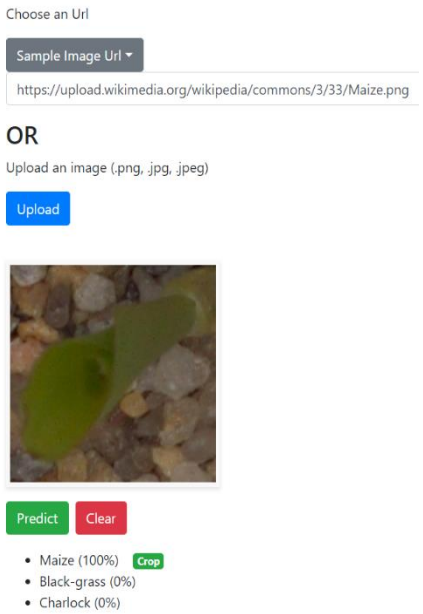


Fig. 21: Demo web application

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

1) top_classes = get_top_2_classes(), // Get the top elements
2) max_1 = get_top_1_value(), max_2 = get_top_2_value() // Get the values corresponding
3) *if top_classes* = ["*Black – grass*", "Loose silky-bent"] and top_class = "Black-grass" {
difference = max1 - max2 // Calculate the difference
if difference >= 0.8792 then output "Black-grass"
otherwise output "Loose silky-bent"
}

Fig. 22: Algorithm for improving classification

The threshold (0.8782) was obtained by calculating the mean of the differences between the two classes. Here is the new confusion matrix obtained (figure 23).

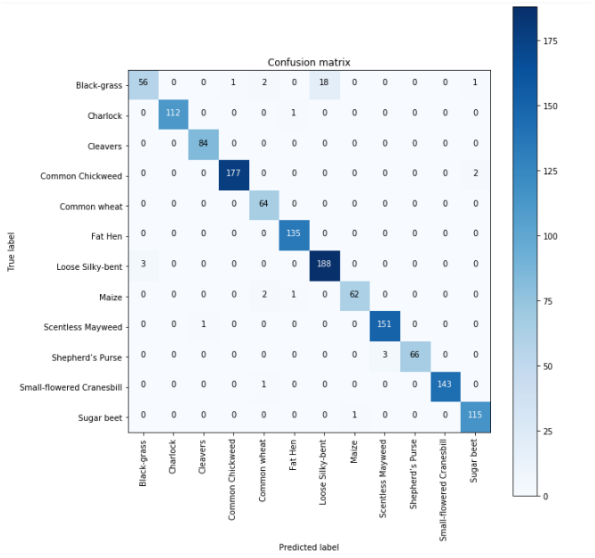


Fig. 23: Confusion matrix for test set with the proposed algorithm

By applying this algorithm on the predictions of our model, the number of wrong predictions reduced, and we increased the accuracy of Black-grass and Loose silky-bent to 98.18% and 98.47% respectively.

VI. CONCLUSION AND FUTURE SCOPE

An accurate deep learning model for crops and weeds classification can definitely assist farmers in maximizing crop yields and consequently minimizing the losses. In this paper, we applied transfer learning technique to classify crops and weeds by using deep convolutional neural networks. The Residual Network 101 architecture was used for conducting this work and we achieved incredible results namely 98.47% of overall accuracy on the validation set and 96.04% on the test set. The dataset contains approximately 5539 images of plants belonging to 12 species at various growth stages. Our proposed approach was the result of several techniques applied successively. Firstly, we converted the dataset images to jpeg format, which reduced the time of processing without impact the accuracy. Then we applied data augmentation techniques such as rotation, zooming, flipping to make more balanced the 12 classes of dataset. After that, we used a pre-trained ResNet 101 model to perform features extraction. We started with the progressive resizing which consists of training a model with a small input image and then using the weights obtained to train the same model with greater images and so on. We applied also the cyclical learning rates method to determine the most appropriate range of learning rates. We observed that the size of input image had a significant impact on the performance of our model. Our model generalizes well with 96.04% of accuracy on the testing set. However, we noticed a misclassification of two particular classes namely Black-grass and Loose Silky-Bent during our experiments. These classes are difficult to distinguish even for humans. That confirms the need of additional data. There exist a lot of plants used in farming. This model can be extended to other species of plants in other countries as well as other online datasets. Our solution is already available online through web application for farmers and can be adapted in future for being used into mobile applications or embedded equipment.

REFERENCES

1. FAO. “How to feed the world in 2050”. http://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf.
2. Univeristy of Pretoria. “Important weeds in maize”. https://www.up.ac.za/sahri/article/1810372/important-weeds-in-maize.
3. Yann LeCun, YoshuaBengio, and Geoffrey Hinton. “Deep learning”. nature, 521(7553):436, 2015.
4. Thomas M. Mitchell. “Machine Learning”. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
5. Saad Abouzahir, Mohamed Sadik, and Essaid Sabir. “Enhanced approach for weeds species detection using machine vision”. *In 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), pages 1-6. IEEE, 2018.*
6. [Jialin Yu, Shaun M Sharpe, Arnold W Schumann, and Nathan S Boyd. “Deep learning for image-based weed detection in turfgrass”. *European Journal of Agronomy, 104:78-84, 2019.*
7. Philipp Lottes, Jens Behley, Andres Milioto, and CyrillStachniss. “Fully convolutional networks with sequential information for robust crop and weed detection in precision farming”. *IEEE Robotics and Automation Letters, 3(4):2870-2877, 2018.*
8. M Dian Bah, Adel Hafiane, and Rapha¨el Canals. “Deep learning with unsupervised data labeling for weeds detection on uav images”. *arXiv preprint arXiv:1805.12395, 2018.*
9. Wenhao Zhang, Mark F Hansen, Timothy N Volonakis, Melvyn Smith, Lyndon Smith, Jim Wilson, Graham Ralston, Laurence Broadbent, and Glynn Wright. “Broad-leaf weed detection in pasture”. *In 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), pages 101-105. IEEE, 2018.*
10. Mads Dyrmann and Rasmus NyholmJørgensen. “Roboweedsupport: weed recognition for reduction of herbicide consumption”. *In Precision agriculture’15, pages 259-269. Wageningen Academic Publishers, 2015.*
11. Catherine R Alimboyong and Alexander A Hernandez. “An improved deep neural network for classification of plant seedling images”. *In 2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA), pages 217-222. IEEE, 2019.*
12. NimaTeimouri, Mads Dyrmann, Per Nielsen, SolvejgMathiassen, Gayle Somerville, and Rasmus Jørgensen. “Weed growth stage estimator using deep convolutional neural networks”. *Sensors, 18(5):1580, 2018.*
13. Alessandro dos Santos Ferreira, Daniel Matte Freitas, GercinaGon¸calves da Silva, HemersonPistori, and Marcelo TheophiloFolhes. “Weed detection in soybean crops using convnets”. *Computers and Electronics in Agriculture, 143:314-324, 2017.*
14. Daniel K Nkemelu, Daniel Omeiza, and Nancy Lubalo. “Deep convolutional neural network for plant seedlings classification”. *arXiv preprint arXiv:1811.08404, 2018.*
15. Sue Han Lee, Chee Seng Chan, Simon Joseph Mayo, and Paolo Remagnino. “How deep learning extracts and learns leaf features for plant classification”. *Pattern Recognition, 71:1-13, 2017.*
16. Andreas Kamilaris and Francesc X Prenafeta-Boldu. “Deep learning in agriculture: A survey”. *Computers and Electronics in Agriculture, 147:70-90,2018.*
17. Konstantinos Liakos, PatriziaBusato, DimitriosMoshou, Simon Pearson, and DionysisBochtis. “Machine learning in agriculture: A review”. Sensors, 18(8):2674, 2018.
18. Gong Cheng, Zhenpeng Li, Junwei Han, Xiwen Yao, and Lei Guo. “Exploring hierarchical convolutional features for hyperspectral image classification”. *IEEE Transactions on Geoscience and Remote Sensing, (99):1-11,2018.*

19. JinruXue, Sigfredo Fuentes, Carlos Poblete-Echeverria, Claudia Gonzalez Viejo, Eden Tongson, Hejuan Du, and BaofengSu. "Automated Chinese medicinal plants classification based on machine learning using leaf morphocolorimetry, fractal dimension and." *International Journal of Agricultural and Biological Engineering", 12(2):123-131, 2019.*
20. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". *In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770-778, 2016.*
21. Sepp Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02):107-116, 1998.*
22. kisspng. Convolutional neural network, kernel, filter, angle, symmetry png. https://www.kisspng.com/png-convolutional-neural-network-kernel-filter-convolu-2326955/.
23. Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. "Learning activation functions to improve deep neural networks". *arXiv preprint arXiv:1412.6830, 2014.*
24. Dipanjan (DJ) Sarkar. A comprehensive hands-on guide to transfer learning with real-world applications in deep learning. https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.
25. SinnoJialin Pan and Qiang Yang. "A survey on transfer learning". *IEEE Transactions on knowledge and data engineering, 22(10):1345-1359, 2010.*
26. Samuel Dodge and Lina Karam. "Understanding how image quality affects deep neural networks". *In 2016 eighth international conference on quality of multimedia experience (QoMEX), pages 1-6. IEEE, 2016.*
27. Wikipedia contributors. Jpeg | Wikipedia, the free encyclopedia, 2019. https://en.wikipedia.org/w/index.php?title=JPEG&oldid=892990014.
28. Uber Engineering. Faster neural networks straight from jpeg. https://eng.uber.com/neural-networks-jpeg/.
29. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". *In 2009 IEEE conference on computer vision and pattern recognition, pages 248{255. Ieee, 2009.*
30. Cadene. "Pretrained convnets for pytorch: Nasnet, resnext, resnet, inceptionv4, inceptionresnetv2, xception, dpn, etc." https://github.com/Cadene/pretrained-models.pytorch#torchvision.
31. Aleksey Bilogur. Boost your cnn image classifier performance with progressive resizing in keras. https://towardsdatascience.com/boost-your-cnn-image-classifier-performance-with-progressive-resizing-in-keras-a7d96
32. Leslie N Smith. "Cyclical learning rates for training neural networks". *In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 464-472. IEEE, 2017.*
33. Daniel Godoy. Understanding binary cross-entropy / log loss: a visual explanation. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.
34. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Doll'ar. "Focal loss for dense object detection". *In Proceedings of the IEEE international conference on computer vision, pages 2980-2988, 2017.*
35. Thomas MosgaardGiselsson, Mads Dyrmann, Rasmus NyholmJørgensen, Peter Kryger Jensen, and Henrik SkovMidtiby. A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. *arXiv preprint, 2017.*
36. Fastai. Fastai: Making neural nets uncool again. https://www.fast.ai/.
37. [37] PyTorch. An open source deep learning platform that provides a seamless path from research prototyping to production deployment. https://www.pytorch.org

AUTHORS PROFILE

Abdel-Aziz BINGUITCHA-FARE is apost graduate student (M.Tech) in computer science and engineering at APG Shimla University, where his area of research revolves around artificial intelligence as well as cloudcomputing. He graduated from University of Lomé (especially from CIC), with a bachelor's degree in computerNetworking. His interest areas are deep learning, devops, cloud computing and software engineering. He is also interested by productivity techniques and entrepreneurship. He also launched a startup in Togo focused on orientation in education domain. Currently, he is looking for a PhD on cloud computing where he could innovate and develop new concepts for providing next-generation solutions.

Prince SHARMA is working as Assistant Professor at Alakh Prakash Goyal Shimla University.He is pursuing the current research from Jaypee University of Information Technology,Solan, Himachal Pradesh. He has worked at Central Potato Research Institute for the Bioinformatics Project. He has guided research in Bioinformatics, Machine Learning, Computer Algorithms for Network in Master students and several Bachelor Projects.He is the member of various International & National professional & academic bodies. He is an active participant of various Seminars, Induction andfaculty development programs

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)