Weakly-supervised learning approach for potato defects segmentation

Sofia Marino, Pierre Beauseroy, André Smolarz

Version of Record: https://www.sciencedirect.com/science/article/pii/S0952197619301630
Manuscript_52874daf487a8d0c502f9616bb27440f

Weakly-supervised learning approach for potato defects segmentation

Sofia Marino*, Pierre Beauseroy, André Smolarz

Institut Charles Delaunay/M2S, FRE 2019, University of Technology of Troyes, 12, rue Marie Curie CS 42060 - 10004, Troyes Cedex, France

Abstract

Rigorous quality analysis of potatoes is essential to define their market price. Manual approaches to detect skin defects of this tuber are laborious, subjective and time-consuming. In this paper, we introduce a weakly-supervised learning method to classify, localize and segment potato defects to automate the quality control task. A large and diversified image-level labeled dataset is created including potatoes from six different classes: healthy, damaged, greening, black dot, common scab and black scurf. A convolutional neural network (CNN) is trained to achieve the classification task. Then, we leverage the discriminative regions that appear in the activation maps of the trained CNN to localize the classified defect. A coarse-to-fine segmentation method is proposed to obtain a more precise defect size. Based on this segmentation, a classification according to the severity of the defect is done, showing the importance of the segmentation phase. Experimental results demonstrate that CNN outperforms conventional classifiers. At a final stage, a multi-label multi-class dataset is used to evaluate the whole system, achieving an average precision of 0.91 and an average recall of 0.90.

Keywords: Weakly-supervised segmentation, Convolutional neural networks, Potato classification, Disease detection, Defect detection, Agricultural applications

1. Introduction

The potato is a famous tubercle cultivated in more than 150 countries around the world[1]. Several defects are prone to appear on the skin of potatoes that affect their quality, and consequently, their sale price. Meticulous quality control is compulsory to define the correct selling price and the market to which the crop will be addressed. Most of the existing methods are based on manual control, in which human operators use the naked eye to observe and classify each tuber. However, this manual task is laborious, subjective and time-consuming, producing classification errors that could be avoided. Therefore, the development of methods that automate quality control is of paramount importance to increase efficiency, reduce costs and obtain objective results that strengthen confidence among customers.

A large number of computer vision and machine learning techniques have been applied to classify and localize defects in various agricultural produce. The first works were focused on the extraction of hand-crafted features

*Corresponding author.
Email addresses: sofia.marino@utt.fr (Sofia Marino), pierre.beauseroy@utt.fr (Pierre Beauseroy), andre.smolarz@utt.fr (André Smolarz)

combined with conventional classifiers. A method was introduced in [2] to detect and classify different external defects in oranges and mandarins using multispectral images and morphological features. A Bayesian discriminant analysis was used to classify citrus between 11 defects, reaching an overall success rate of 86%. Despite the satisfactory results obtained, the authors proposed to use two types of cameras, VIS and NIR, which makes the cost of the vision system higher. A clustering technique was applied in [3] to quantify the damaged in banana hands. Two k-means algorithms were applied: the first one sought to segment banana images from the background, and the second one was applied to assess its lesions and spots. Qualitative results were shown, where the segmented images produced by the proposed method were compared with experts manual segmentation. Although interesting results were reported, the defects were not classified by types. Authors in [4] proposed a method to detect defective tomatoes. Color and texture features extracted from images were used to feed a three layers neural network. This network was trained to classify tomatoes between defective and non-defective. Finally, the non-defective tomatoes were classified as ripe or unripe using only the R, G, B values of the image. A classification accuracy of 100% and 96.47% was reached for the defective/non-defective and ripe/unripe respectively. While the results were satisfactory, the system was able to process 300 images per hour, which may be slow for some industrial applications. Authors in [5] proposed a method for grading golden apples. A dataset with pixel-level annotations was used to train a Multi-Layer Perceptron (MLP) neural network to segment defect. Statistical, textural and geometric features were extracted from the segmented defects to train a classifier that was able to grade the apples. Support Vector Machines (SVM), MLP and K-Nearest Neighbor (KNN) classifiers were evaluated, obtaining the best recognition rate of 92.5% with the SVM classifier. Computer vision techniques applied to potato defects detection were also studied. A pixel-wise AdaBoost classifier was used in [6] to detect 5 different defects in potatoes. A large number of candidate features were extracted to select the most important to train the AdaBoost classifier. They achieved a success rate of 89.6% and 89.5% for white and red potatoes. Even if a small data set of 102 images was enough to achieve great results, the authors showed that expert pixel-level defect annotations were subjective and error-prone, affecting the performance of the system. A method to detect potato defects and to sort them by size is proposed in [7]. A mathematical binarization was used to sort the potatoes while color features were extracted to segment the defects by a SVM classifier. An accuracy of 95% was obtained for the segmentation task and about 96.86% for the size grading phase. As in the method proposed in [6], a pixel-level labeled database was used to perform segmentation, which construction is time-consuming and not error-free. Authors in [8] proposed a method to classify diseases on potato plants. A public dataset was used to train and test their method. The Gray Level Co-occurrence Matrix (GLCM) was used to extract textural features which were combined with color features to train a SVM classifier. Healthy, late blight and early blight images were classified reaching a F1-score of 92%, 95% and 98% respectively.

A major advantage of the above methods, which are mainly based on hand-crafted features, is their ability to obtain satisfactory results even with a small database. However, methods that propose to segment defects usually need pixel-level annotations that are difficult to get. Another drawback is that they largely depend on the right choice of features to be extracted. Normally, these hand-crafted features are adapted to a specific problem and they lack of

generalization.

Recently, various methods based on deep learning have been developed to overcome this issue. Several computer vision tasks, such as image classification [9, 10, 11, 12], object detection [13, 14, 15] and semantic segmentation [16, 17, 18], have been assessed using deep learning-based methods, outperforming traditional methods. The main advantage is that they can automatically find a suitable representation from the raw data to achieve the classification or detection task. Deep learning has been rapidly applied in agriculture, obtaining promising results. Authors in [19] proposed to train an adapted Deep Residual Neural Network to detect three different diseases in plants. A large pixel-level labeled dataset was created, reaching a balanced accuracy of 0.87 for early and late diseases. An advantage of the proposed method is that it can be applied under natural conditions, which is important for a real digital agricultural application. However, great efforts were made to obtain a sufficiently varied and well-labeled database. In [20], the authors applied a deep convolutional neural network (CNN) to classify four diseases in cucumber leaves. Private and public datasets were used to create the database. They compared the results obtained by traditional methods, showing that better performance was achieved by the CNN. Although the proposed method obtained good results, the segmentation of the defect was not indispensable, as they only sought to classify the images by disease, regardless of its severity. In [21], the authors proposed a new CNN architecture based on AlexNet [9] and VGGNet [10] to classify weed and crop species. They achieved high accuracy in the classification task (98.21%), outperforming other pre-trained networks. Authors in [22] proposed a multiple stage method to identify radish wilt disease. Similar to their previous work [23], they combined hand-crafted features and deep learning to classify Fusarium wilt of radish. Unmanned aerial vehicles (UAVs) were used to obtain the dataset. A k-means clustering was applied to divide the input image between three regions: radish, ground and matching film. The number of groups k was selected according to segmentation results, comparing it with a built pixel-level labeled dataset. Finally, the Fusarium wilt of radish was classified according to severity, reaching an accuracy of over 90%.

Although deep learning methods are not widely explored in the classification of potato defects, some works have been presented. A patch-wise classification method was proposed by [24]. They trained a CNN to classify patches extracted from potato images between five classes: healthy, black dot, black scurf, silver scurf and common scab. Experiments using different training datasets were carried out, reaching an accuracy of 95.85%. Working with patches made the database larger without the need for more tubers. However, the method was not developed to perform defect segmentation. Authors in [25] proposed an ensemble-based classifier to detect sprouting in potatoes. Learned features combined with hand-crafted features were used to train traditional classifiers such as SVM, KNN and AdaBoost. They outperformed mainstream methods with a prediction rate of 0.916. However, one drawback of using the ensemble-based classifier is the increase of testing time, which is of crucial importance in some real-world applications.

As shown, excellent results were obtained with the methods based on deep learning in the field of agriculture. Nevertheless, these methods have one major disadvantage: they require a large set of manually labeled data to perform the classification task. Besides, if we seek to perform segmentation of defects, we should commonly use a large dataset with pixel-level labels, which construction is laborious and time-consuming. Image-level labels are much

simpler to get. Thus, many efforts have been made to develop weakly-supervised segmentation methods that can leverage only image-level annotations [26, 27, 28, 29, 30]. Most of these proposed approaches are focused on the object segmentation task, where objects with well-defined contours are segmented. An instance weakly-supervised segmentation method was introduced in [30] that used peak response maps to mask instances in images. In addition, object proposals off-the-shelf methods were applied to perform a coarse-to-fine segmentation. Although great results were found, the method is focused on object segmentation. In our application, results could be adversely affected because some defects could be scattered over the entire surface of the potato. Authors in [31] proposed a weakly-supervised method to count fruits in images. Two losses were combined: a Presence-Absence classifier (PAC) to detect images which contains at least one fruit and a spatial consistency loss that impose coherency among counting results obtained at different levels. They showed comparable results to fully-supervised methods with the advantage of using only image-level labels.

In this paper, we propose to tackle the potato defect segmentation problem in a weakly-supervised manner, where the defect spots may not be well-defined, resulting in a more challenging problem. The segmentation of defects is crucial to classify them according to their severity, which depends, among other factors, on the surface area of the affected potato. Therefore, working in a weakly-supervised manner minimizes the work of image labeling since we only need image-level annotations. In this way, we avoid creating a pixel-level labeled database that requires a lot of time, effort and human expertise. Also, there are certain defects in potatoes, such as *black dot*, which are very difficult to identify and could affect the quality of the database if pixel-level annotations were needed.

The main contributions of this work are as follows:

- A convolutional neural network is trained with a large dataset to classify potatoes into six different classes: healthy, damaged, greening, black dot, common scab and black scurf.
- A Defect Activation Map (DAM) is generated to localize the classified defect in the image.
- A coarse-to-fine segmentation method is introduced to obtain the actual size of the defect.
- Finally, the segmentation results are used as input to SVM classifiers to classify damaged and greening potatoes by gravity.

The rest of the paper is organized as follows: the proposed method is explained in detail in Section 2. Discussion and results are presented in Section 3. Finally, we conclude the paper in Section 4.

2. Materials and Methods

In this section, we first introduce our dataset. Then we describe the proposed method with a detailed description of each phase.

2.1. Potato images

Potatoes from six distinct classes, i.e., healthy, damaged, greening, black dot, common scab and black scurf, were used to create the dataset. A digital camera was positioned in a black box and a LED illumination source was used. The camera took 4 images of different sides of each potato, placed on a black background. The resolution of the images was 1602×882. Several potato varieties such as Agata, Monalisa, Gourmandine, Annabelle, Caesar, Charlotte and Marilyn, were included to generate a dataset as varied as possible. A total of 2422 tubers were available, resulting in a final dataset of 9688 images. Some of these images are shown in Figure 1. In order to train and test the algorithm, two experts manually annotated all images in two different ways: at first, each face image was classified individually between 6 different classes. As shown in Table 1, the face-wise dataset was divided in 5325 healthy, 984 damaged, 1263 greening, 597 black dot, 1276 common scab and 243 black scurf images. Secondly, a potato-wise classification was done, taking into account the 4 sides images coming from the same tuber. In this setting, damaged and greening potatoes were divided by gravity, resulting in a final 8 classes dataset. The decision to divide only damaged and greening images by gravity was due to sample availability. As depicted in Table 2, the number of images in each class, i.e. healthy, light damaged, serious damaged, light greening, serious greening, black dot, common scab and black scurf, was 831, 341, 159, 161, 349, 151, 359 and 71 respectively. Due to a the large number of images to be classified, it is important to note that no information about the location of the defect (as bounding boxes or pixel-level annotations) was used, which simplified the task for the experts. The dataset was then divided using 70% of potato images for training and validate the models, and 30% for testing. In this last test dataset, we could have more than one class per potato (including all 4 sides together), resulting in a multi-label multi-class dataset.

Table 1: Face-wise image classification dataset.

Class	Number of images
Healthy	5325
Damaged	984
Greening	1263
Black dot	597
Common scab	1276
Black scurf	243
Total	9688

2.2. Proposed method

The global scheme of the proposed method, as presented in Figure 2, consists of four phases. Firstly, a CNN was trained to classify each side of the potato into six different classes. Secondly, the detected defect was localized (not apply for the healthy class), showing the key regions of the image that incited the CNN to make the decision. In the

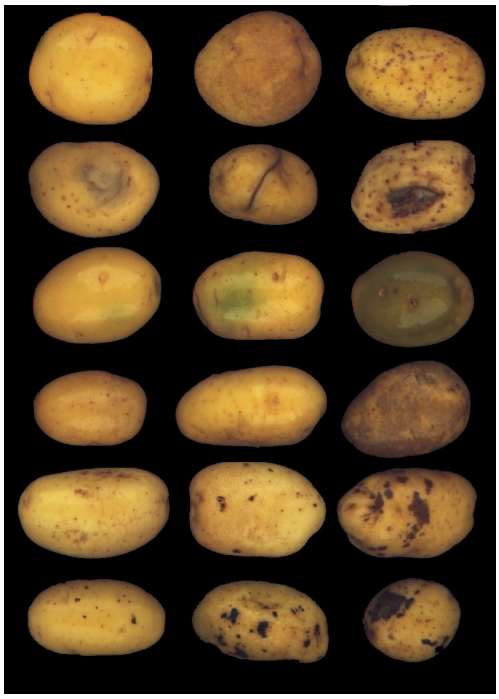

Figure 1: Images of the six classes with variable gravity. By rows, from top to bottom: healthy, damaged, greening, black dot, common scab and black scurf.

Table 2: Potato-wise image classification dataset.

Class	Number of images
Healthy	831
Light damaged	341
Serious damaged	159
Light greening	161
Serious greening	349
Black dot	151
Common scab	359
Black scurf	71
Total	2422

third phase, a coarse-to-fine segmentation method was introduced, to obtain an exact location of the defect. Finally, the segmentation results were used to train two SVMs to classify damaged and greening potatoes by gravity. We give a detailed explanation of each phase in the following sections.

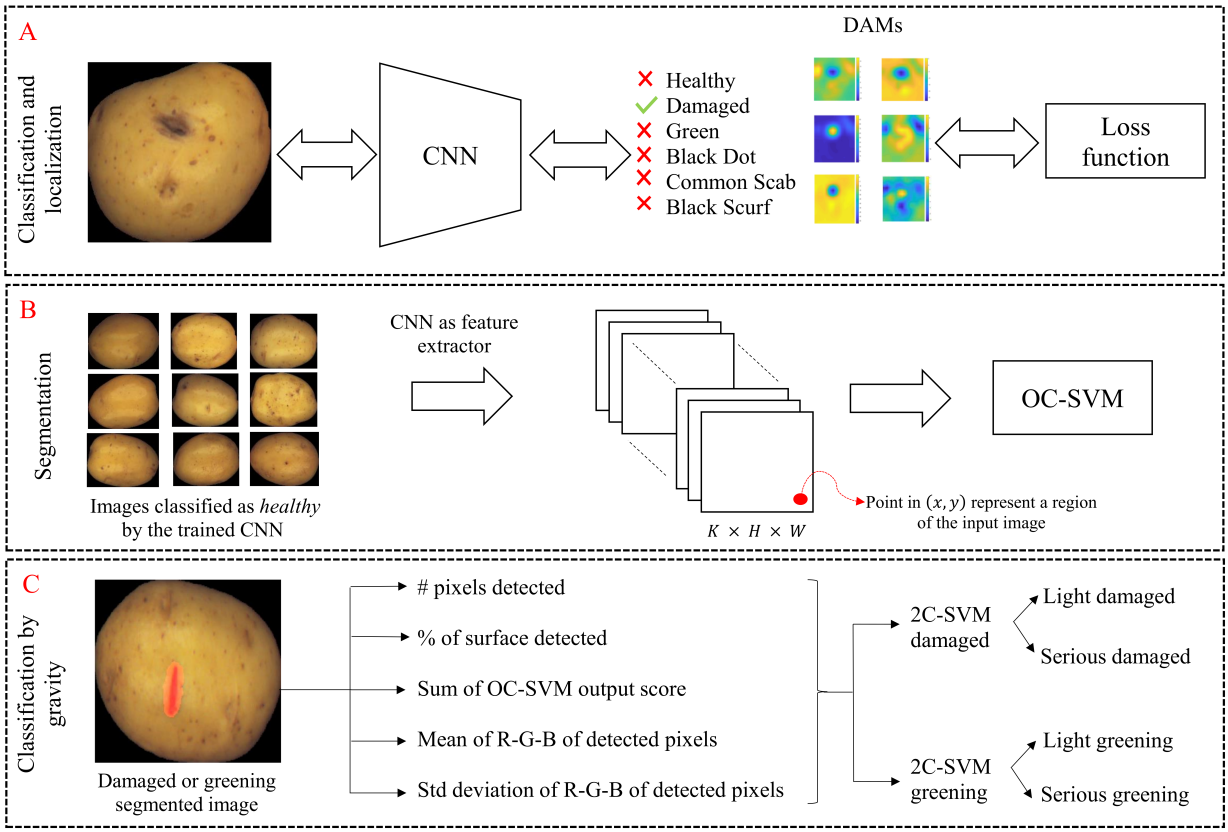

Figure 2: Scheme of the proposed method.

2.2.1. Face-wise image classification by CNN

To accomplish the first phase, we fine-tuned a pre-trained CNN to classify each side potato image into six distinct classes (healthy, damaged, greening, black dot, common scab and black scurf). Three famous architectures were evaluated to compare their performance in our specific task (AlexNet[9], VGG-16[10] and GoogLeNet[11]). All networks were initially trained with the ImageNet[32] dataset to classify images into one of 1000 categories. To fine-tune the three networks, we replaced their last fully-connected layer by a new one adapted to the number of potato classes and a softmax activation function was used. After modifying the networks, the final number of learning parameters was 58.3M, 134.2M and 5.5M for AlexNet, VGG-16 and GoogLeNet, respectively.

The training dataset consisted of only one class per image, which justify the softmax function applied in the final classification layer. Nevertheless, in a real-world application, the same face image may have more than one defect. For this reason we proposed to apply in the prediction step a sigmoid function together with the softmax, where the class outputs are independent of one another. Given an image I, the activation map of unit k of the last convolutional layer is represented by $f_k(x, y)$, for $k = 1, 2, \ldots, K$, where K is the total number of filters in the convolutional layer. Applying a global average pooling (GAP) to each activation map k give us $F_k = \sum_{x,y} f_k(x, y)$, $F_k \in \mathbb{R}$, which followed

by the last fully-connected layer generates the defect score S_d for a given class d:

$$S_d = \sum_{k=1}^{K} w_k^d F_k \tag{1}$$

where w_k^d are the weights learned in the last fully-connected layer for the class d at unit k. The defect score S_d of each class d is used as input to the softmax (Eq.2) and sigmoid (Eq.3) functions.

$$P_d = \frac{e^{S_d}}{\sum_d e^{S_d}} \tag{2}$$

$$Q_d = \frac{1}{1 + e^{-S_d}} \tag{3}$$

To decide which class or classes the image belong to, the following decision rule was applied:

1. First predicted class:

$$\max_d P_d \tag{4}$$

2. Other possible class:

$$\text{if} \quad Q_d \geq h_{sigmoid} \;\; \forall d \in D \tag{5}$$

$$\text{d = possible predicted class}$$

where $h_{sigmoid}$ is a threshold experimentally selected to decide if the input image I belongs to the class d.

2.2.2. Defect localization

In the second phase, we localized the defects classified by the CNN. Inspired by the work proposed in [29], we generated a Defect Activation Map (DAM) that shows the localization of the classified defect. The DAM was generated using a weighted sum of the activation maps of the last convolutional layer with the weights learned in the last fully-connected layer. In the CNN, the first layers represent more general information, and deeper layers are focused on more complex and discriminant patterns specific to the classification task. Therefore, the activation maps of the last convolutional layer preserve the spatial location of specific patterns found in the input image. In contrast to the weakly-supervised method proposed by [30], the DAM is generated by a single forward, which makes this method more appropriate for obtaining quick results at the inference stage.

Given an image I, the defect score S_d of each class d is computed by Eq. 1. The weights learned in the last fully-connected layer (w_k^d) used to compute S_d measure the contribution of each activation map of unit k, $f_k(x, y)$, to the predicted class d. Thus, the DAM is generated by applying a weighted sum of the activation maps as follows:

$$DAM_d(x, y) = \sum_{k=1}^{K} w_k^d f_k(x, y) \tag{6}$$

A bilinear up-sampling was applied to each DAM to match the size of the input image where we could identify the key regions that incited the CNN to predict a certain class. In the prediction phase, we also used the Defect Activation Maps to decide if the image could belong to more than one class. If there is a second potential class d predicted by Eq. 5, we define if it is possible to accept it according to the overlap of DAMs, i.e., we calculate the intersection between the DAMs of the output classes and we verify if there is no intersection between them. If the overlap of the DAMs is less than 20%, both classes are kept. Otherwise, only the softmax output class is retained.

In this phase we also evaluated a modified version of the GoogLeNet architecture. The modified network, called GoogLeNet_Modif, is a less deep network where the spatial resolution of the last convolutional layer is higher. Increasing the spatial resolution could lead to an improvement in the precision of the defect localization[29]. For the GoogLeNet_Modif, we only kept layers until the sixth inception module of the original GoogLeNet, increasing the resolution of the last activation map from 7×7 to 14×14. Then, we added three new layers: a convolutional layer with 1024 filters of size 3×3, padding 1 and stride 1, a global average pooling and a fully-connected layer with a softmax activation.

Although the localization results could give an idea of the spatial location of the defects, a finer segmentation is mandatory to obtain exploitable results to classify defects by gravity.

2.2.3. Coarse-to-fine defect segmentation

In the third phase, we aimed to obtain a coarse-to-fine segmentation of the classified defect. Since a database with pixel-level annotations was not available, the network could not be trained end-to-end to obtain defects segmentation. Our proposed method did not use any prior information about the location or extent of the defect but leveraged the features learned by the trained CNN. That is, once the CNN was trained with potato images, it was used as a feature extractor. From images classified as healthy by the trained network, we created a new database with features coming from the lth convolutional layer. Since the CNN had detected these images as healthy, it could be considered that all features extracted from them were *normal*. The objective was to find *abnormal* samples, which include all types of defects.

The output of a convolutional layer l with K filters is of size $K \times H \times W$. An input image is then described by $(H \times W)$ feature vectors of dimension K. Thus, each feature vector represents a specific region of the input image. A One-Class Support Vector Machine (OC-SVM)[33] was trained with the $(n_h \times H \times W)$ normal samples of dimension K, where n_h is the number of predicted healthy images. An important advantage of this classifier is that it only needs normal samples, which are straightforward to obtain.

To adapt the SVM algorithm to non-linear problems, we used a kernel function $k(\cdot, \cdot)$ which is described as:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \tag{7}$$

where $x \in \chi$ is the input data and $\phi(x)$ is the mapping function that project the input data χ to a new feature space $\mathcal{H}$. The objective of the OC-SVM is to find a hyperplane that separates the normal samples from the origin with

maximum margin, by solving the following optimization problem:

$$\min_{w,\xi,\rho}\left(\frac{1}{2}\|w\|^2+\frac{1}{\nu n_h}\sum_{i=1}^{n_h}\xi_i-\rho\right) \tag{8}$$

subject to:

$$(w\cdot\phi(x_i))\geq\rho-\xi_i,\quad \forall i=1,...,n_h \tag{9}$$

$$\xi_i\geq 0,\quad \forall i=1,...,n_h \tag{10}$$

where w and ρ are the learning parameters, $\nu\in(0,1)$ is an upper bound on the fraction of outliers and a lower bound of the fraction of support vectors and ξ_i a slack variable. To decide whether an example x_i belongs to the normal or abnormal class, we compute the following decision function:

$$y_i=\text{sign}\left((w\cdot\phi(x_i))-\rho\right) \tag{11}$$

where $y_i\in\{-1,1\}$ indicates the output class of the example x_i.

At the prediction step, we used the trained OC-SVM classifier to distinguish the abnormal pixels within the Defect Activation Map to obtain a finer segmentation of the defect. This gives us the essential input information to the next phase, where we classify damaged and greening potatoes by gravity.

2.2.4. Potato-wise classification by gravity

In the last phase, we sought to classify damaged and greening potato images by gravity: serious or light. The results obtained in Section 2.2.3 were used as input features to train two SVM[34] classifiers:

- Number of pixels detected as abnormal.
- Percentage of surface detected as abnormal $\left(\frac{\text{Number abnormal pixels}}{\text{Number of total pixels}}\right)$.
- Sum of OC-SVM output score of detected abnormal pixels.
- Mean of R, G and B channels of detected abnormal pixels.
- Standard deviation of R, G and B channels of detected abnormal pixels.

To classify these defects by gravity, we took into account the four faces of the same potato. Thus, to characterize the whole potato we only retained the segmentation results of the face where the biggest defect was detected. This decision was made for two main reasons: firstly, we had available only a potato-wise dataset labeled by gravity. Secondly, we avoided counting more than once the same defect.

2.3. Performance evaluation

We selected the following evaluation metrics to take into account the imbalanced nature of the dataset[35]:

- Confusion matrix: compare the predicted classes (rows) with the ground-truth class (columns).
- $\text{Precision}_\text{d}$: is the ratio of accurately predicted samples of class d to the total predicted samples of class d (Eq. 12).
- Recall_d: is the ratio of accurately predicted samples of class d to all real class d samples (Eq. 13).
- F1-score_d: is the harmonic mean of precision and recall of class d (Eq. 14).

$$P_d = \frac{TP_d}{TP_d + FP_d} \tag{12}$$

$$R_d = \frac{TP_d}{TP_d + FN_d} \tag{13}$$

$$F1\text{-}score_d = 2 * \frac{P_d * R_d}{P_d + R_d} \tag{14}$$

where TP_d is true positives of class d, FP_d is false positives of class d and FN_d is false negatives of class d.

To compare the performance of the proposed method, conventional approaches based on hand-crafted features were used. Color and textural features were extracted from potato images to use as the input of a SVM classifier. Three color spaces were used: RGB, HSV and CIEL*a*b. The extracted color features consisted of mean, variance, entropy and skew of the nine different channels. The Gray-Level Co-Ocurrence Matrix (GLCM)[36] of each channel was used to extract the texture features including contrast, correlation, energy and homogeneity. All these features were used to characterize the images.

3. Results and discussion

In this section, we present the results obtained in each phase of the proposed method. We also show the comparative results, using hand-crafted features and conventional classifiers. The results attained demonstrate that our proposed method can accurately classify defects in potato images, outperforming conventional methods. We finally expose the importance of a fine segmentation of defects to perform a classification by gravity with satisfactory results. Implementation was made in Matlab®R2017b using the Deep Learning Toolbox™. A machine with Ubuntu 16.04 operating system that uses Intel®Core™i7-7700HQ processor, 16GB of DDR4 RAM and a GPU NVIDIA GeForce®1050Ti (4 GB memory) was used for experiments.

3.1. Face-wise image classification

We fine-tuned three CNN architectures: AlexNet, VGG-16 and GoogLeNet. Input images were resized to a specific input size, depending on the architecture of the network that we used (227×227 for AlexNet and 224×224 for VGG-16 and GoogLeNet). To increase the variability and the number of samples used to train the CNN, data augmentation techniques such as rotation and horizontal and vertical flipping were randomly applied. Stochastic gradient descent with batch size of 10, momentum of 0.9 and weight decay of $1x10^{-4}$ was used for training the networks. The maximum number of epochs was set to 100. The learning rate was set to $1x10^{-4}$ for all layers except for the added fully-connected layer, which was set to 20 times the initial learning rate. We applied k-fold cross validation (with $k = 5$) to obtain the results of Table 3. We can observe that the best F1-score for all classes was obtained with GoogLeNet architecture, reaching an average F1-score of 0.94 against to 0.92 and 0.88 for AlexNet and VGG-16 respectively. The greening class had the best F1-score with all architectures: 0.96, 0.95 and 0.98 with AlexNet, VGG-16 and GoogLeNet respectively. The lowest performance was achieved on the black dot class with the three different architectures. This can be explained due to the similarity between this class and the healthy class. To ensure that this sort of confusion was the cause of the lowest performance, we analyzed the confusion matrices, shown in Table 4. As expected, the black dot class was mainly confused with the healthy class, which also occur in the real world, when human operators classify tubers manually.

We compared the obtained results with methods based on hand-crafted features. We used the 72 features extracted from images to train a SVM classifier. Cross-validation of k-fold ($k = 5$) was also applied. Experiments were performed using the original imbalanced dataset ($SVM_{Imbalanced}$) and a balanced version ($SVM_{Balanced}$). To obtain the latter, an undersampling technique was applied ending up with 174 images per class, which is the number of training images of the minority class. In this way, each class had the same number of samples, resulting in a balanced database to train the SVM. Furthermore, to handle the imbalance dataset without eliminating samples from majority classes (as it was done to train the $SVM_{Balanced}$), we adjusted the class-weights when training the SVM, which we refer in this work as $SVM_{Weighted}$. Following the work of authors in [37], we adjusted the misclassification penalty parameter (C) of the SVM by affecting it with different weighting factors for each class. Given two classes of size n_1 and n_2, we define the weighting factors of each class, a_1 and a_2, with $a_1 + a_2 = 1$, by the following equation:

$$\frac{a_1}{a_2} = \frac{n_2}{n_1} \tag{15}$$

We can observe from Eq. 15 that the minority class will have a higher weighting factor. The aim is to compensate for the adverse effects caused by the unequal size of training classes.

The comparative results are shown in Table 5, where we can observe that CNNs outperformed conventional classifiers, increasing the average F1-score from 0.78 (obtained with $SVM_{Imbalanced}$ classifier) to 0.94 (obtained with GoogLeNet). Detailed results using hand-crafted features are shown in Table 6. As expected, the performance on the healthy class was affected when using the balanced dataset, which was the majority class in the original dataset.

Table 3: Precision, recall and F1-score obtained after fine-tuning the three different CNN architectures. Classes are: H=Healthy, D=Damaged, G=Greening, BD=Black Dot, CS=Common Scab and BS=Black Scurf.

Metric	Class	AlexNet	VGG-16	GoogLeNet
Precision	H	0.93 ± 0.03	0.91 ± 0.02	**0.95 ± 0.02**
	D	0.94 ± 0.03	0.93 ± 0.02	**0.96 ± 0.02**
	G	**0.99 ± 0.01**	0.98 ± 0.01	0.98 ± 0.01
	BD	0.86 ± 0.05	0.86 ± 0.08	**0.93 ± 0.06**
	CS	0.96 ± 0.02	0.94 ± 0.03	**0.97 ± 0.02**
	BS	0.91 ± 0.06	0.85 ± 0.03	**0.92 ± 0.04**
Recall	H	**0.98 ± 0.01**	**0.98 ± 0.01**	0.98 ± 0.02
	D	0.89 ± 0.05	0.84 ± 0.04	**0.92 ± 0.02**
	G	0.93 ± 0.06	0.92 ± 0.02	**0.97 ± 0.01**
	BD	**0.79 ± 0.04**	0.68 ± 0.10	0.78 ± 0.06
	CS	0.90 ± 0.06	0.86 ± 0.02	**0.95 ± 0.02**
	BS	0.96 ± 0.03	0.86 ± 0.07	**0.97 ± 0.04**
F1-score	H	0.95 ± 0.02	0.94 ± 0.01	**0.97 ± 0.01**
	D	0.92 ± 0.03	0.88 ± 0.03	**0.94 ± 0.02**
	G	0.96 ± 0.04	0.95 ± 0.01	**0.98 ± 0.02**
	BD	0.82 ± 0.04	0.76 ± 0.03	**0.85 ± 0.06**
	CS	0.93 ± 0.03	0.90 ± 0.01	**0.96 ± 0.02**
	BS	0.93 ± 0.04	0.86 ± 0.04	**0.95 ± 0.04**

The lowest results were obtained with the undersampling technique ($SVM_{Balanced}$). This can be explained because in this setting several samples were not used to train the classifier, which impairs its performance. Similar to the results obtained with the CNN, the lowest recall was obtained on the black dot, which was mostly confused with the healthy class, as depicted in the confusion matrices of Table 7.

Due to the outstanding results obtained with the GoogLeNet architecture, we decided to continue the experiments (localization and segmentation of defects) using this architecture.

3.2. Defect localization

The localization results were evaluated qualitatively by the Defect Activation Maps (DAM). A threshold technique was applied to the DAM to obtain a segmented heatmap. We kept all DAM values greater or equal to a specific percentage of the maximum DAM value. The final threshold of 0.4 was chosen experimentally. We compared the GoogLeNet with the GoogLeNet_Modif to verify if the increase of the activation map resolution had a positive impact in the localization results. Figure 3 shows the DAM with different thresholds applied. As expected, a more accurate

Table 4: Confusion matrix obtained after fine-tuning AlexNet, VGG-16 and GoogLeNet. Classes are: H=Healthy, D=Damaged, G=Greening, BD=Black Dot, CS=Common Scab and BS=Black Scurf. Values in %.

		Ground-Truth(%)					
	Class	H	D	G	BD	CS	BS
Pred.(%)	AlexNet						
	H	97.5	6.6	5.9	20.0	7.9	2.9
	D	0.8	89.4	0.1	0.2	0.5	0.0
	G	0.0	0.6	93.3	0.2	0.0	0.0
	BD	1.2	0.7	0.7	78.8	0.2	0.0
	CS	0.4	2.2	0.0	0.7	90.2	1.2
	BS	0.1	0.4	0.0	0.0	1.2	96.0
Pred.(%)	VGG-16						
	H	97.6	11.5	7.0	28.8	9.6	8.1
	D	0.6	83.7	0.1	0.7	1.1	2.3
	G	0.3	0.4	91.9	0.7	0.1	0.0
	BD	0.7	1.3	0.7	68.4	1.2	0.0
	CS	0.7	2.4	0.2	1.2	86.2	3.5
	BS	0.1	0.6	0.1	0.2	1.7	86.2
Pred.(%)	GoogLeNet						
	H	98.1	6.3	2.8	20.5	3.7	1.1
	D	0.6	92.4	0.0	0.2	0.3	0.6
	G	0.2	0.1	96.9	0.5	0.1	0.0
	BD	0.6	0.0	0.2	78.4	0.2	0.0
	CS	0.4	1.0	0.1	0.5	94.5	1.1
	BS	0.1	0.1	0.0	0.0	1.1	97.1

Table 5: Average precision, average recall and average F1-score obtained with all methods.

Method	Avg Precision	Avg Recall	Avg F1-score
AlexNet	0.93	0.91	0.92
VGG-16	0.91	0.86	0.88
GoogLeNet	**0.95**	**0.93**	**0.94**
$SVM_{Balanced}$	0.66	0.76	0.69
$SVM_{Weighted}$	0.75	0.78	0.76
$SVM_{Imbalanced}$	0.83	0.74	0.78

Table 6: Precision, recall and F1-score obtained with hand-crafted features and SVM classifier. Classes are: H=Healthy, D=Damaged, G=Greening, BD=Black Dot, CS=Common Scab and BS=Black Scurf.

Metric	Class	$SVM_{Balanced}$	$SVM_{Weighted}$	$SVM_{Imbalanced}$
Precision	H	**0.91 ± 0.01**	0.90 ± 0.01	0.85 ± 0.01
	D	0.46 ± 0.05	0.57 ± 0.02	**0.74 ± 0.06**
	G	0.74 ± 0.02	0.89 ± 0.02	**0.93 ± 0.01**
	BD	0.40 ± 0.04	0.51 ± 0.03	**0.69 ± 0.08**
	CS	0.83 ± 0.03	0.83 ± 0.02	**0.86 ± 0.03**
	BS	0.61 ± 0.06	0.81 ± 0.07	**0.90 ± 0.07**
Recall	H	0.71 ± 0.01	0.84 ± 0.02	**0.94 ± 0.01**
	D	**0.72 ± 0.03**	0.70 ± 0.02	0.51 ± 0.08
	G	0.88 ± 0.03	**0.89 ± 0.03**	**0.89 ± 0.02**
	BD	**0.63 ± 0.03**	0.61 ± 0.03	0.48 ± 0.06
	CS	0.77 ± 0.01	**0.85 ± 0.04**	**0.85 ± 0.04**
	BS	**0.86 ± 0.02**	0.77 ± 0.06	0.78 ± 0.08
F1-score	H	0.80 ± 0.01	0.87 ± 0.01	**0.89 ± 0.01**
	D	0.56 ± 0.03	**0.63 ± 0.01**	0.60 ± 0.06
	G	0.81 ± 0.01	0.89 ± 0.02	**0.91 ± 0.01**
	BD	0.49 ± 0.03	0.55 ± 0.03	**0.56 ± 0.07**
	CS	0.80 ± 0.01	0.84 ± 0.01	**0.85 ± 0.02**
	BS	0.71 ± 0.04	0.79 ± 0.02	**0.83 ± 0.04**

localization result was achieved with GoogLeNet_Modif network. On the other hand, GoogLeNet attained better results in the face-wise classification task, as shown in Table 8.

3.3. Defect segmentation

We extracted the features from the 8th layer of the trained CNN, resulting in a feature map of size $56 \times 56 \times 192$. This feature map can be interpreted as 56×56 feature vectors of 192 dimensions. Features from images classified as healthy by the trained network were used to train a OC-SVM_{Seg} with a Gaussian kernel. An example of the OC-SVM_{Seg} output is shown in Figure 4. An up-sampling operation was made to the heatmap segmentation output, to match the size of the input image. For each new test image, if it was not detected as healthy, a Defect Activation Map was computed in order to localize the defect. Then, all pixels within the DAM were tested by the OC-SVM_{Seg} classifier to decide if the pixel was a defect or not. Figure 5 shows an example of the results obtained with the coarse-to-fine segmentation phase.

Due the fact that we did not have a segmented labeled dataset, we also evaluated the performance of the segmenta-

Table 7: Confusion matrix obtained with hand-crafted features and SVM classifier. Classes are: H=Healthy, D=Damaged, G=Greening, BD=Black Dot, CS=Common Scab and BS=Black Scurf. Values in %.

		Ground-Truth(%)					
	Class	H	D	G	BD	CS	BS
	$SVM_{Balanced}$						
Pred.(%)	H	71.4	11.2	6.7	15.8	6.2	3.5
	D	12.6	71.6	2.6	8.4	4.1	2.3
	G	5.7	3.9	88.5	5.1	0.9	0.6
	BD	7.6	8.2	1.7	62.6	5.6	2.3
	CS	2.3	3.5	0.4	4.6	77.4	5.7
	BS	0.4	1.6	0.1	3.5	5.8	85.6
	$SVM_{Weighted}$						
Pred.(%)	H	84.2	17.2	7.1	23.0	6.2	1.2
	D	7.5	70.1	1.6	6.3	3.7	3.5
	G	1.8	2.1	88.9	2.8	0.5	0.0
	BD	4.5	5.2	2.1	60.9	3.3	1.1
	CS	1.8	4.5	0.2	6.3	84.9	17.2
	BS	0.2	0.9	0.1	0.7	1.4	77.0
	$SVM_{Imbalanced}$						
Pred.(%)	H	94.5	40.1	10.2	39.8	10.2	5.2
	D	1.9	50.5	0.5	3.5	2.4	3.4
	G	0.9	1.6	88.6	2.3	0.3	0.0
	BD	1.4	3.0	0.6	47.9	1.4	0.6
	CS	1.2	4.2	0.1	6.5	84.9	13.2
	BS	0.1	0.6	0.0	0.0	0.8	77.6

Table 8: Average precision, average recall and average F1-score obtained after fine-tuning GoogLeNet and GoogLeNet_Modif

Metric	GoogLeNet	GoogLeNet_Modif
Average Precision	**0.95**	0.90
Average Recall	**0.93**	0.89
Average F1-score	**0.94**	0.89

tion method by the last classification phase, where we used the segmentation results to classify damaged and greening potato images by gravity.

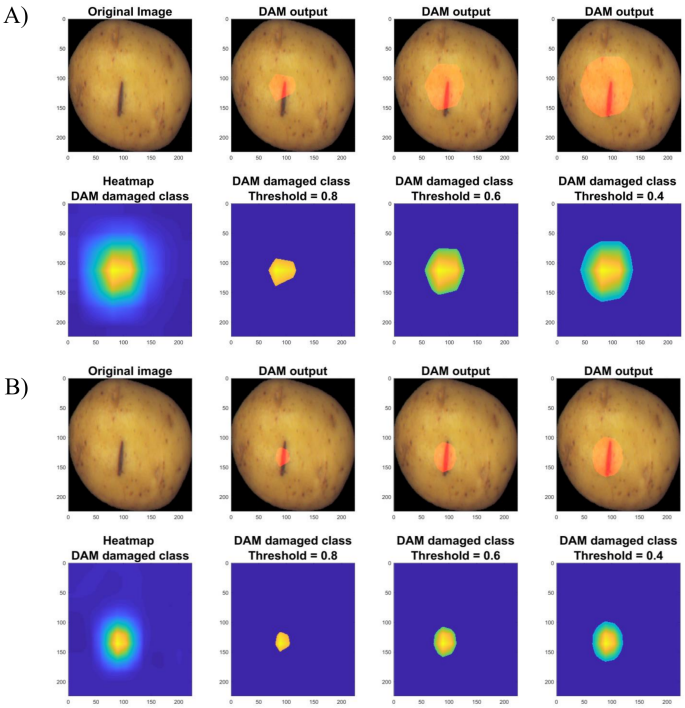

Figure 3: Example of DAM generated with different threshold values. By column: original image, threshold = 0.8, threshold = 0.6 and threshold = 0.4. A) GoogLeNet and B) GoogLeNet_Modif.

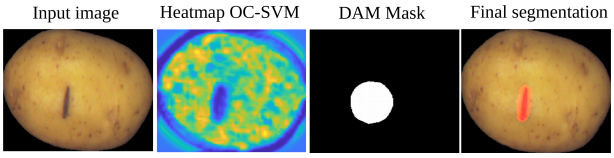

Figure 4: Example of intermediate outputs until the final result. By column: input image, heatmap output of OC-SVM$_{\text{Seg}}$, thresholded Defect Activation Map (threshold = 0.4) and final segmented image.

3.4. Classification by gravity: light and serious defect

In the final phase, we classified damaged and greening images by gravity. Only healthy, damaged and greening classified potatoes were taken into account to train and validate the models. Until this phase, each side of the same potato was classified separately (4 different sides per potato), but to classify defect gravity, we considered the four images of each potato together. This decision was made because we only had available a potato-wise dataset labeled by gravity. We retained the segmentation results of the side where the biggest defect was detected. For example, if two sides of the same potato were classified as greening, we only use the segmentation results from the face where we have localized the biggest green spot.

The segmentation results were used as input of two SVM$_{\text{Gravity}}$ classifiers to differentiate between light and serious defects. Potatoes images were finally classified between Light Damaged (LD) or Serious Damaged (SD) and Light

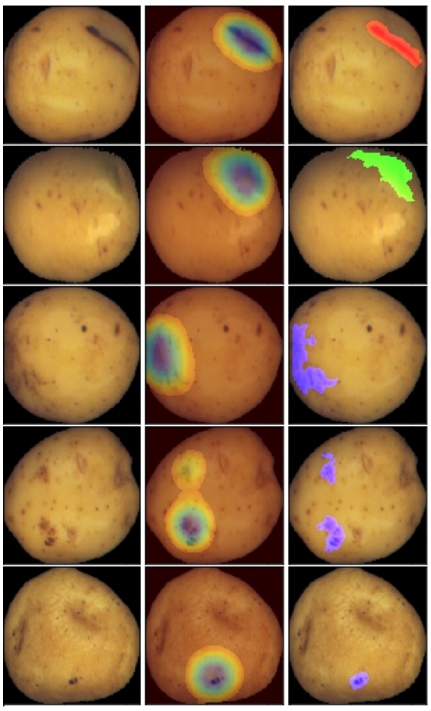

Figure 5: Localization and segmentation examples on the test set. By row different classes: damaged, greening, black dot, common scab and black scurf. It can be seen that the Defect Activation Maps (second column) gives the localization of the classified defect using a threshold of 0.4, which is then segmented in a finer way by the features extracted with the CNN and the OC-SVM$_{\text{Seg}}$ (third column).

Greening (LG) or Serious Greening (SG).

In order to be able to evaluate the importance of the segmentation phase, we analyzed the scenario where we only used the localization results to classify images by gravity, i.e., without applying the coarse-to-fine segmentation phase. Cross-validation and grid search were applied to select the SVM$_{\text{Gravity}}$ hyperparameters, $\sigma_{Gaussian}$ and C. Confusion matrices of Table 9 and Table 10 show the results obtained for damaged and greening potatoes respectively. As we can see, GoogLeNet_Modif was more accurate to differentiate between light and serious defects, which confirms that the localization and segmentation made by this network was better. We can also observe that the coarse-to-fine segmentation phase is mandatory to be able to differentiate potatoes by gravity. The confusion between the serious damaged and the light damaged class was more than twice lower when we do a finer segmentation of the defect (15.5% versus 37.3%). The same occurred with light greening and serious greening classes, reducing its confusion from 50.5% to 15.1%.

3.5. Global evaluation of the tuber

A multi-label multi-class test set containing 722 tubers was available to test the whole system. We took into account the four output labels obtained in the previous stages, one per each face image, to characterize the whole potato. Because in the testing set exist images where more than one class is present at the same time, we applied the decision

Table 9: Confusion matrix using GoogLeNet+OC-SVM$_{Seg}$+SVM$_{Gravity}$ (GNet), GoogLeNet_Modif+OC-SVM$_{Seg}$+SVM$_{Gravity}$ (GNet_Modif) and GoogLeNet+SVM$_{Gravity}$ without the coarse-to-fine segmentation phase (W/O segmentation) to classify potato images between H=Healthy, LD=Light damaged, SD=Serious damaged.

		Ground-Truth(%)		
	Class	H	LD	SD
	GNet			
Pred.(%)	H	99.3	6.7	0.9
Pred.(%)	LD	0.7	89.1	15.5
Pred.(%)	SD	0.0	4.2	83.6
	GNet_Modif			
Pred.(%)	H	99.7	10.5	1.8
Pred.(%)	LD	0.3	87.4	10.0
Pred.(%)	SD	0.0	2.1	88.2
	W/O segmentation			
Pred.(%)	H	99.3	6.7	0.9
Pred.(%)	LD	0.7	85.7	37.3
Pred.(%)	SD	0.0	7.6	61.8

rule explained in Section 2.2.1 and the overlapping analysis of DAMs explained in Section 2.2.2 to decide the final classification results (see example Figure 6). Three different architectures were tested to obtain the final results: GoogLeNet, GoogLeNet_Modif and a combination of both networks, where we classified with GoogLeNet and we localized and segmented defects with GoogLeNet_Modif. Results are shown in Table 11. We can observe that the whole system can accurately classify the potatoes. The best performance was achieved when combining both networks, reaching an average F1-score of 0.90, against to 0.88 and 0.89 for GoogLeNet and GoogLeNet_Modif respectively. The healthy class attained the best performance with a F1-score of 0.98. The class black dot had the smallest F1-score (0.86 with the combination setting) due to the confusion with healthy images. We can also observe that we can precisely classify damaged and greening potatoes by gravity, leveraging the localization and segmentation results.

According to the results obtained, the proposed method can be used to carry out a precise and efficient quality analysis. Besides, the results obtained are objective, avoiding the uncertainty factor that operators have when performing quality control manually. Finally, this system could be applied in the detection of defects in various fruits and vegetables, only needing to have a database with image-level annotations.

Table 10: Confusion matrix using GoogLeNet+OC-SVM$_{Seg}$+SVM$_{Gravity}$ (GNet), GoogLeNet_Modif+OC-SVM$_{Seg}$+SVM$_{Gravity}$ (GNet_Modif) and GoogLeNet+SVM$_{Gravity}$ without the coarse-to-fine segmentation phase (W/O segmentation) to classify potato images between H=Healthy, LG=Light greening, SG=Serious greening.

		Ground-Truth(%)		
	Class	H	LG	SG
	GNet			
Pred.(%)	H	100	3.2	0.0
Pred.(%)	LG	0.0	81.7	6.1
Pred.(%)	SG	0.0	15.1	93.9
	GNet_Modif			
Pred.(%)	H	99.8	2.1	0.0
Pred.(%)	LG	0.2	83.9	5.3
Pred.(%)	SG	0.0	14.0	94.7
	W/O segmentation			
Pred.(%)	H	100	3.3	0.0
Pred.(%)	LG	0.0	46.2	7.2
Pred.(%)	SG	0.0	50.5	92.8

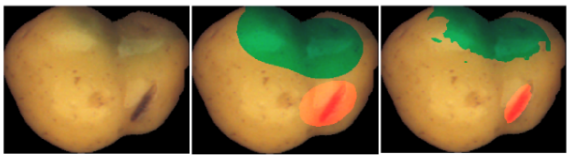

Figure 6: Example of a face potato image with two defects: damaged and greening. By column: original image, DAMs output, segmentation output.

4. Conclusion

In this work, we propose a weakly-supervised deep learning method to classify, localize and segment defects on potatoes. An image-level dataset has been created, dividing potatoes into 6 classes: healthy, damaged, greening, black dot, common scab and black scurf. Firstly we have compared the performance of three CNN architectures to classify images by defect. Secondly, a Defect Activation Map (DAM) generated by the trained CNN has been used to localize the defect regions. A coarse-to-fine segmentation method has been applied in the third phase to obtain the actual size of the defect. No prior information about the location or extent of the defect has been needed to achieve the segmentation. Finally, in the fourth phase, the damaged and greening defect segmentation results have been used as input to a SVM classifier which has been trained to distinguish these defects by gravity: light or serious.

Table 11: Precision, recall and F1-score of testing set using three different frameworks: GoogLeNet+OC-SVM$_{Seg}$+SVM$_{Gravity}$ (GNet), GoogLeNet_Modif+OC-SVM$_{Seg}$+SVM$_{Gravity}$(GNet_Modif) and a combination of both networks (Combination). Classes are: H=Healthy, LD=Light Damaged, SD=Serious Damaged, LG=Light Greening, SG=Serious Greening, BD=Black Dot, CS=Common Scab and BS=Black Scurf.

Metric	Classes	GNet	GNet_Modif	Combination
Precision	H	**0.98**	0.95	**0.98**
	LD	0.83	**0.88**	0.85
	SD	0.79	0.84	**0.91**
	LG	0.89	0.83	**0.93**
	SG	0.90	0.92	**0.94**
	BD	**0.92**	0.91	0.90
	CS	0.95	**0.96**	0.94
	BS	**0.84**	0.78	0.81
	Average	0.89	0.88	**0.91**
Recall	H	0.98	**0.99**	0.98
	LD	0.91	0.87	**0.95**
	SD	0.77	**0.88**	0.83
	LG	0.70	**0.85**	0.83
	SG	0.96	**0.97**	**0.97**
	BD	**0.82**	0.75	**0.82**
	CS	0.88	0.88	0.88
	BS	0.95	0.95	0.95
	Average	0.87	0.89	**0.90**
F1-score	H	**0.98**	0.97	**0.98**
	LD	0.86	0.88	**0.90**
	SD	0.78	0.86	**0.87**
	LG	0.78	0.84	**0.87**
	SG	0.93	0.95	**0.96**
	BD	**0.87**	0.82	0.86
	CS	0.91	0.91	0.91
	BS	**0.89**	0.86	0.88
	Average	0.88	0.89	**0.90**

Experimental results showed that the CNN achieved an excellent classification performance, reaching an average F1-score of 0.94. We demonstrated that this method outperformed conventional approaches based on hand-crafted features. We also exposed the need for precise segmentation of defects to be able to differentiate potatoes defects by

gravity. A reduction of more than a half in the confusion between light and serious defects was achieved by using the DAM together with the segmentation method (from 37.3% to 15.5% in damaged class and from 50.5% to 15.1% in the greening class). The final global evaluation on a multi-label multi-class dataset reached an average F1-score of 0.90 combining two different CNN architectures.

Our proposed weakly-supervised defect segmentation method can be used in an industrial application due to its high efficiency and speed in the inference stage (0.417 sec/img). Besides, it only requires an image-level labeled dataset, which is less time-consuming and much more abundant than accurate segmented datasets. It is important to note that no smoothing priors are added to achieve the segmentation task, as bounding box candidates or superpixels, which is difficult in our case due to the lack of well-defined shapes of certain defects.

Future works consist of experimenting with multispectral cameras. Although there would be an increase in the cost of the system, a method based on multispectral images could improve the results obtained. In particular, we could improve the detection of the black dot defect, which has had the lowest performance. Furthermore, we could also perform a detection of internal defects in tubercles, which is unfeasible with RGB images. We will also consider improvements of the acquisition system to guarantee 100% coverage of the surface area of potatoes. When working with 2D images, the defects found on the extremes cannot be seen. The use of 3D images will be explored to be able to analyze the entire surface of the potato at once, including the extremes. Finally, the current campaigns should make possible to enhance the database. This will further improve the analysis of defects that could not be classified by gravity (*black dot, common scab* and *black scurf*) due to the unavailability of a database.

References

References

[1] Faostat, `http://www.fao.org`, accessed: 2019-05-23.

[2] J. Blasco, N. Aleixos, J. Gómez-Sanchis, E. Moltó, Recognition and classification of external skin damage in citrus fruits using multispectral data and morphological features, Biosystems engineering 103 (2) (2009) 137–145.

[3] M.-h. Hu, Q.-l. Dong, B.-l. Liu, P. K. Malakar, The potential of double k-means clustering for banana image segmentation, Journal of Food Process Engineering 37 (1) (2014) 10–18.

[4] M. P. Arakeri, et al., Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry, Procedia Computer Science 79 (2016) 426–433.

[5] P. Moallem, A. Serajoddin, H. Pourghassem, Computer vision-based apple grading for golden delicious apples based on surface features, Information Processing in Agriculture 4 (1) (2017) 33–40.

[6] M. Barnes, T. Duckett, G. Cielniak, G. Stroud, G. Harper, Visual detection of blemishes in potatoes using minimalist boosted classifiers, Journal of Food Engineering 98 (3) (2010) 339–346.

[7] N. Razmjooy, B. S. Mousavi, F. Soleymani, A real-time mathematical computer method for potato inspection using machine vision, Computers & Mathematics with Applications 63 (1) (2012) 268–279.

[8] M. Islam, A. Dinh, K. Wahid, P. Bhowmik, Detection of potato diseases using image segmentation and multiclass support vector machine, in: Electrical and Computer Engineering (CCECE), 2017 IEEE 30th Canadian Conference on, IEEE, 2017, pp. 1–4.

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

[10] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[12] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, arXiv preprint arXiv:1704.06904 (2017).

[13] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[14] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in neural information processing systems, 2016, pp. 379–387.

[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[16] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: Multi-path refinement networks for high-resolution semantic segmentation, in: IEEE conference on computer vision and pattern recognition (CVPR), Vol. 1, 2017, p. 3.

[17] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2881–2890.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE transactions on pattern analysis and machine intelligence 40 (4) (2018) 834–848.

[19] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, A. Johannes, Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild, Computers and Electronics in Agriculture (2018).

[20] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, Z. Sun, A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network, Computers and Electronics in Agriculture 154 (2018) 18–24.

[21] T. R. Chavan, A. V. Nandedkar, Agroavnet for crops and weeds classification: A step forward in automatic farming, Computers and Electronics in Agriculture 154 (2018) 361–372.

[22] L. M. Dang, S. I. Hassan, I. Suhyeon, A. kumar Sangaiah, I. Mehmood, S. Rho, S. Seo, H. Moon, Uav based wilt detection system via convolutional neural networks, Sustainable Computing: Informatics and Systems (2018).

[23] J. G. Ha, H. Moon, J. T. Kwak, S. I. Hassan, M. Dang, O. N. Lee, H. Y. Park, Deep convolutional neural network for classifying fusarium wilt of radish from unmanned aerial vehicles, Journal of Applied Remote Sensing 11 (4) (2017) 042621.

[24] D. Oppenheim, G. Shani, Potato disease classification using convolution neural networks, Advances in Animal Biosciences 8 (2) (2017) 244–249.

[25] W. Ming, J. Du, D. Shen, Z. Zhang, X. Li, J. R. Ma, F. Wang, J. Ma, Visual detection of sprouting in potatoes using ensemble-based classifier, Journal of Food Process Engineering 41 (3) (2018) e12667.

[26] P. O. Pinheiro, R. Collobert, From image-level to pixel-level labeling with convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1713–1721.

[27] G. Papandreou, L.-C. Chen, K. P. Murphy, A. L. Yuille, Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1742–1750.

[28] A. Kolesnikov, C. H. Lampert, Seed, expand and constrain: Three principles for weakly-supervised image segmentation, in: European Conference on Computer Vision, Springer, 2016, pp. 695–711.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[30] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, J. Jiao, Weakly supervised instance segmentation using class peak response, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3791–3800.

[31] E. Bellocchio, T. A. Ciarfuglia, G. Costante, P. Valigi, Weakly supervised fruit counting for yield estimation using spatial consistency, IEEE Robotics and Automation Letters (2019).

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[33] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, Support vector method for novelty detection, in: Advances in neural information processing systems, 2000, pp. 582–588.

[34] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[35] M. Bekkar, H. K. Djemaa, T. A. Alitouche, Evaluation measures for models assessment over imbalanced datasets, Iournal Of Information Engineering and Applications 3 (10) (2013).

[36] R. M. Haralick, K. Shanmugam, et al., Textural features for image classification, IEEE Transactions on systems, man, and cybernetics (6) (1973) 610–621.

[37] Y.-M. Huang, S.-X. Du, Weighted support vector machine for classification with uneven training class sizes, in: 2005 International Conference on Machine Learning and Cybernetics, Vol. 7, IEEE, 2005, pp. 4365–4369.