Fruit Classification for Retail Stores using Deep Learning

Deleted for blind review

No Institute Given

Abstract. Payment of fruits or vegetables in retail stores normally require them to be manually identified. This paper presents an image classification method, based on lightweight Convolutional Neural Networks (CNN), with the goal of speeding up the checkout process in stores. A new dataset of images is introduced that considers three classes of fruits, inside or without plastic bags. In order to increase the classification accuracy, different input features are added into the CNN architecture. Such inputs are, a single RGB color, the RGB histogram, and the RGB centroid obtained from K-means clustering. The results show an overall 95% classification accuracy for fruits with no plastic bag, and 93% for fruits in a plastic bag.

Keywords: Deep learning · Convolutional neural networks · Fruit classification.

1 Introduction

Retail stores rely on cashiers or self-service checkout systems to process the customers' purchases. Since most products have barcodes that can be scanned, the checkout time has already been minimized. However, fruits and vegetables are commonly processed differently. The cashier or the customer need to manually identify the class of product being bought and look for it in the system. The purpose of this paper is to present an image classification method with the goal of accelerating the checkout process of such products.

Fruit classification is a complex problem due to all the variations that can be encountered. In general, two classification problems can be identified: i) classification of fruits of different types (e.g., to differentiate between oranges and apples) [3], and ii) classification of varieties of the same fruit (e.g., to differentiate among apple varieties such as, red delicious, honeycrisp, golden delicious, gala, etc.) [8]. However, even focusing on the first type of problem, precise classification is still difficult to achieve due to differences in shape, color, ripening stages, etc. Another problem, directly related to the purchase of fruits in retail stores, is that fruits can be inside a plastic bag. This research work deals with the first type of classification (i.e., classification of fruits of different types), where fruits can be inside or without a plastic bag.

Recent advancements in Convolutional Neural Networks (CNN) [17] make them suitable for this problem. The ImageNet Large Scale Visual Recognition

Challenge (ILSVRC)[13] was an annual competition that started in 2010 and ended in 2017[1]. Given an image, the goal was to recognize the different objects within such image. Since 2012, CNN have had an outstanding performance in the task of image classification, because the winner of ILSVRC used a model based on deep CNN trained on raw RGB pixel values, known as AlexNet [10]. Besides AlexNet, several CNN architectures have been defined throughout the years, such as LeNet, ZFNet, GoogleNet, VGG, ResNet, YOLO, MobileNetV2, among others [17]. In 2015, ResNet was the winner of ILSVRC exceeding for the first time the human-level accuracy (5% error) [4,13] with a 3% of testing error.

This paper proposes an improved CNN architecture based on MobileNetV2 [15] to classify fruits, where it proposes the addition of different input features (besides the input images), in order to improve its accuracy. Such additional inputs features are related to the color of the fruits. Thus, we present experiments using a single RGB color, the RGB histogram, and the RGB centroid obtained from K-means clustering. In addition, we created a new fruit dataset for three types of fruits: apples, oranges and bananas; also considering fruits in transparent plastic bags. The results show an overall 95% classification accuracy for fruits without plastic bag, and 93% for fruits inside a plastic bag.

The paper is is organized as follows. Section 2 summarizes related work on fruit classification. Section 3 describes the proposed classification method. Section 4 presents some experiments and analyzes the results of the classification performance. Finally, Section 5 discusses the conclusions and future work.

2 Related Work

There are several research works for fruit recognition and classification with different goals and applications. One of this applications refers to agriculture and fruit harvesting. *DeepFruits* is a Faster Region-based CNN (known as R-CNN). Their model employs transfer learning using ImageNet, and two types of input images: color (RGB) and Near-Infrared (NIR). The images correspond to seven fruits still attached to their corresponding tree/plant, so this application is oriented to agricultural robots for harvesting fruit and vegetables. The images were taken by some of the authors and others obtained from Google Image searches [14]. *Deep Count* is another application for robotic agriculture using a Deep Neural Network (DNN), where the authors propose a modified Inception-ResNet architecture. Their research only focuses on tomato images from Google Images [12]. Another related application is *Deep Fruit Detection* for robotic harvesting in orchards. That research employs Faster R-CNN and compares the performance against other architectures such as VGG and ZFNet. They also explore the number of training images, transfer learning and data augmentation. They study three fruits: apple, mango and almond; with RGB images generated by themselves [2]. Finally, *MangoYOLO* is also a CNN model but for mango harvest forecast. *MangoYOLO* is compared with other CNN architectures including Faster R-CNN with VGG and ZFNet, SDD and YOLO.

[1] http://image-net.org/

Their research explores the number of training images and transfer learning with PASCAL VOC, COCO and ImageNet. They obtained their own RGB images at night with a special LED system mounted on a farm vehicle to obtain consistent illumination conditions [9]. The above research works have to consider other factors such as working outdoors, variations in lighting conditions and the fact that fruits/vegetables are still attached to the trees.

Besides agriculture, retail applications can greatly benefit from the classification of fruits and vegetables. Hossain et al. [5], proposed two CNN architectures, a light model of six CNN layers, and a VGG-16 fine-tuned model. They also created their own dataset by collecting images from the Internet. Another research work proposed a double-track method using a two nine-layer CNN [8]. The input of the first network are images with background, then the second network works with a single fruit selected from a region of interest. Rather than classifying types of fruits, they classified six varieties of apples. Finally, Femling et al. [3], describes a hardware system for retail stores able to classify ten types of fruits. They make use of a dataset comprised of images from ImageNet and taken with their system's camera. As in our case, they made use of CNN architecture based on MobileNet. It is important to note that the works just reviewed do not consider the fruits to be inside plastic bags, as this paper does.

3 Application of the Proposed Method

This section presents our proposed method for solving the fruit classification problem. First, a new dataset is introduced, then we explain the chosen CNN architectures and training methods.

3.1 Dataset

Data is an essential part of Deep Learning. Therefore, it is important to select the correct input data according to our goals. For fruit classification, there exists a dataset called Fruits-360 [11], which consists of 28736 training images and 9673 testing images. It contains 60 different classes of fruits, where some classes refer to varieties of a fruit (e.g., for apples they have six varieties). A major drawback of this dataset is that the images are small (100×100 pixels), which makes it difficult to differentiate between some fruits. Also, the images have no background, thus it does not scale very well to real-world applications.

Therefore, we decided to create our own dataset. Since our main goal is to replicate a store environment, the fruits were placed over a stainless steel sheet and the photos were taken from the top. We chose to work with three types of fruit: apples, oranges and bananas. We introduce variation in the dataset, by taking images of the fruits at different positions and rotations (see Figure 1). We also consider that in the checkout process the fruits are generally inside a transparent plastic bag, so photos of the fruits in a bag were also taken. The photos were taken using the front camera of an iPhone 6. In total, 1067 images were collected, 725 for training and 342 for testing. Table 1 shows the number of images for training and testing per category.

Fig. 1: Examples of training images of the dataset created.

Table 1: Number of images per category in the dataset created.

Fruit Name	Training images	Testing images	Total images
Apple	297	146	443
Orange	242	121	363
Banana	156	75	231

3.2 Selecting a CNN Architecture

The selection of a CNN architecture depends on the problem one is trying to solve. There is no unique architecture for all problems. Thus, selecting the right one becomes a problem on its own. Currently the top performing architecture, that won the ILSVRC competition in 2017, [13] is the *Ensamble C* developed by the *WMW team* which uses an Squeeze-and-Excitation block that adaptively recalibrates channel-wise features [7], but it has the disadvantage of being computational expensive to train and for making predictions. On the other hand, MobileNets are CNN architectures that deal with the computational complexity problem by implementing a more efficient and lightweight architecture, and could run on mobile or embedded devices achieving high-level performance. This is done by dividing a normal convolution into two steps. First performing a point-wise convolution, then a depth-wise convolution [6]. MobileNetV2 uses inverted residual blocks and bottleneck layers to achieve better performance [15]. Since our goal is to work in retail stores, we have chosen to use the MobileNetV2 architecture for being lightweight and robust.

3.3 Transfer Learning

Transfer learning is a machine learning approach where a model developed for one task becomes the starting point for a model of a different task [16]. This technique works really well when the available dataset is not large enough, and also the model converges faster. Therefore, we trained MobileNetV2 with transfer learning using weights from a model trained with the ImageNet dataset.

There are several ways to train a model using transfer learning. For our work, we first loaded the pre-trained model and discarded the last layer. This is a dense layer with 1000 neurons that serves as a classifier of the previous feature map. Once discarded, we set the rest of the layers to *not trainable* (this prevents the weights in a given layer from being updated). Then, at the end of the network we added another dense layer, but now with the number of classes of fruits we want to predict. This allows to keep all the features extracted from the ImageNet model, and re-purpose it to the fruit classification problem. Then, we trained the model for 20 epochs using the base learning rate of $1e^{-4}$. After the first 20 epochs, we set the layers from the 100 layer up to the last layer (155) to *trainable*, and we trained the network for another 20 epochs. But this time, we set the learning to 1/10 of the base learning rate in order to *fine-tune* the model. This forces the weights to be tuned from generic feature maps to features associated specifically with our dataset.

In this work, the models are trained using TensorFlow [1], with the implementation of MobileNetV2 provided by Keras. The standard RMSPropOptimizer is used, with both, decay and momentum set to 0.9. We use batch normalization after every layer, where the standard weight decay is set to 0.00004 as described in [15]. The base learning rate is set to $1e^{-4}$ and a batch size set to 50. The models were trained using an iMac with a 3.5 GHz Intel Core i5, and 8 GB of RAM 1600 MHz DDR3. Figure 2 shows a preliminary comparison of the models performance when trained with transfer learning and with weights initialized randomly, using our dataset. It is clear that the results prove that with random weights the model is not able to learn, while the model trained with transfer learning reaches almost 0.80 accuracy.

3.4 Improving MobileNetV2

One way to visualize what a CNN model is learning is by looking at the convolutional layers activations, in order to see what information is being retained by the layers. Figure 3 shows the activations, of the first convolutional layer, of two different fruits. At the top row images show activations of an orange, while at the bottom row are activations of an apple. As can be noticed, for the model both fruits are similar. It mainly retains the shape of the fruit and its texture. This information might not be enough to differentiate between both fruits due to their similar shape. One missing feature that, in this case, would be important for such differentiation is the color of the fruits. Therefore, the accuracy of the model can be improved if additional input features (related to the fruit's color) are feed into the model. This work proposes three different input features and their corresponding modifications to the model.

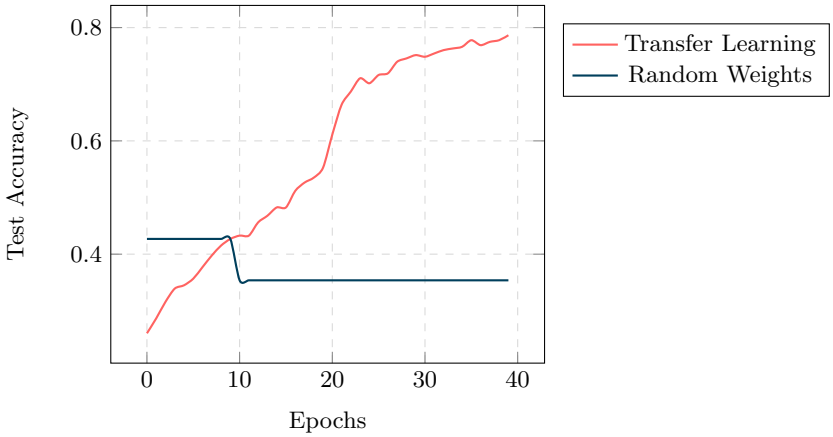


Fig. 2: Comparing the test accuracy using our dataset and transfer learning

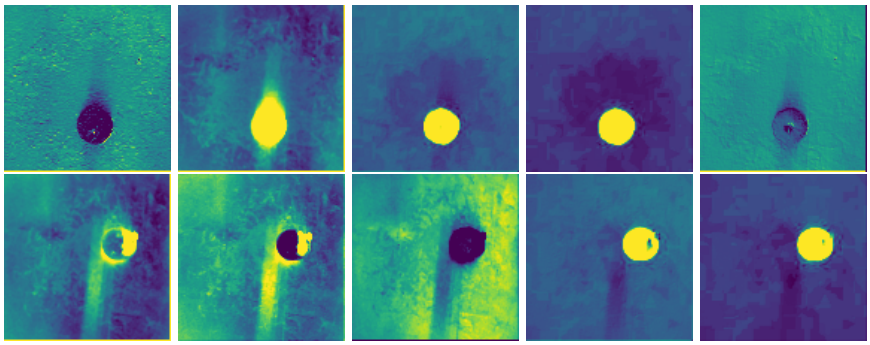


Fig. 3: Similar activations of the first convolutional layer of an orange (top image) compared to an apple (bottom image)

Single RGB Fruit Color Besides the image, one additional input feature is to provide the model with a vector with RGB color values of the fruit to be classified. This color should be the one that represents, in general, the given fruit. For instance, bananas would be represented by the yellow color, thus the model receives the vector with RGB color values $[1.0, 1.0, 0.0]$; in the case of an orange the vector would be $[1.0, 0.64, 0.0]$. This vector with three RGB values is feed into the model.

RGB Histogram An image histogram is a graph that summarizes how many pixels are at different scale levels of a given image [18]. For this work, the histogram of each RGB channel was obtained, resulting in a vector of 765 input values, which are then feed into the model. Figure 4 shows an example of an

image RGB histogram. At this moment, one disadvantage is that most of the values would correspond to the background colors.

RGB Centroid using K-Means Finally, we make use of a hybrid machine learning (ML) approach. The idea is to combine different ML algorithms in order to complement each other. K-Means is a clustering algorithm that tries to partition the data into K clusters (subgroups). When applied to an image, it could find groups of colors that represent such image. For this work, we set the number of groups to three, as shown in the Figure 5. The three RGB colors found (9 values) are fed into the model.

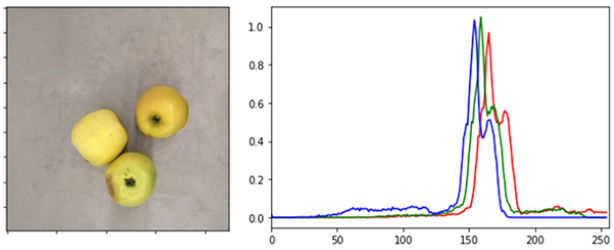


Fig. 4: Example of an image RGB histogram

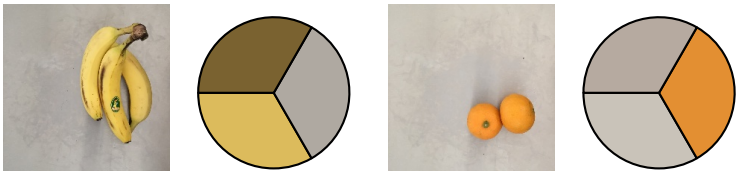
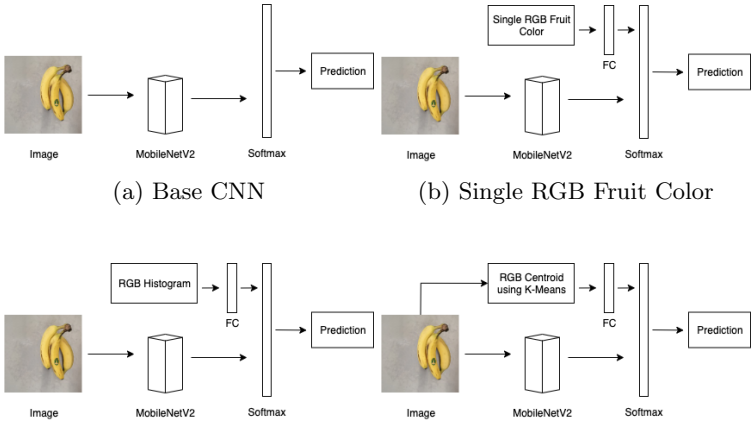


Fig. 5: Example of the RGB Centroid using K-Means

In the case of the RGB color and RGB histogram, a multi-input model is implemented. The model takes as input the image and a vector with the color data, as shown in Figures 6b and 6c. The image is feed into the CNN (i.e., the MobileNetV2 architecture), while the color data is feed into a dense layer. The result of both networks is then concatenated and the final prediction is made using *softmax* activation. Using K-Means, the model is considered a hybrid model. We implemented the K-Means algorithm in TensorFlow as a Keras layer, this allows the model to internally produce the K colors, and then, these are concatenated at the end of the process (as shown in Figure 6d). For our experiments, we chose to use three centroids ($k = 3$), resulting in 9 RGB values. In the following section, we compare how the models perform using the methods just explained.



(a) Base CNN (b) Single RGB Fruit Color

(c) RGB Histogram (d) RGB Centroid using K-Means

Fig. 6: Different architectures of the proposed methods

4 Experiments and Results

As explained in the previous section, our models are based on the MobileNetV2 architecture and trained in two versions of the dataset we created: i) images with only fruits (no bags), and ii) images with fruits without and inside plastic bags.

Table 2 compares the accuracy of the baseline model (MobileNetV2), the multi-input models (both, the single RGB color and the RGB histogram), and the hybrid model (MobileNetV2 + K-Means). In all cases, the accuracy is higher when no plastic bag is used. This is expected since the use of plastic bags distort the look of the fruits. The baseline model (MobileNetV2), with no additional color information, has high accuracy on the training set, but has the lowest on the testing set. Meanwhile, all three models that use additional color information achieved better accuracy in general. Particularly, the model using the single RGB color obtained the highest accuracy at 0.95 and 0.93, for both versions of the dataset, respectively. The relatively lower accuracy achieved by the hybrid model could be due to the fact that, out of three colors obtained, only one is related to the color of the fruit. The other two colors are related to the background, which should be not be taken into account. It is very likely that this same problem is also happening to the RGB histogram. Therefore, one course of future work is to try to eliminate as much background information as possible. Figure 7 compares the testing data accuracy over time of the different proposed methods in this paper, using the complete dataset.

Table 2: Comparing the accuracy of the trained models.

	In Plastic Bag		No Plastic Bag	
Model	Train accuracy	Test accuracy	Train accuracy	Test accuracy
MobileNetV2	0.98	0.78	0.99	0.82
MobileNetV2 + Single color	0.98	**0.93**	0.99	**0.95**
MobileNetV2 + Histogram	0.99	0.82	0.99	0.92
MobileNetV2 + K-Means	0.98	0.86	0.99	0.90

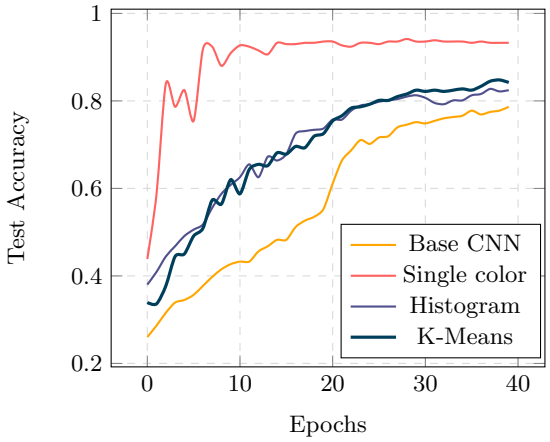


Fig. 7: Accuracy of the models trained with images containing fruits in bags.

5 Conclusions and Future Work

This paper proposed an improved CNN architecture, based on a lightweight CNN architecture called MobileNetV2, by considering additional input features, besides the input images. Such input features improved the accuracy of the model by adding information about the color of the fruits. These input features are: i) the single RGB fruit color, ii) the RGB histogram, and iii) the RGB centroid using K-Means. The single RGB color achieved the best overall accuracy: 95% classification accuracy for fruits with no plastic bag, and 93% for fruits in a plastic bag. Due to the lack of data, a new dataset was introduced consisting of 725 images for training and 342 for testing; and considers three classes of fruits (apples, oranges and bananas). The dataset also considers fruits inside plastic bags. As further research, we are exploring the minimum number of training images to achieve the highest accuracy. Along with this, data augmentation has been not explored using our proposed dataset. We also would like to measure the sensitivity to illumination. On the other hand, we plan to compare the accuracy of the proposed lightweight CNN architecture against other state-of-the-art CNN networks with GPU hardware and our dataset.

References

1. Abadi, M., Agarwal, A., Barham, P., Goodfellow, I., et. al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Bargoti, S., Underwood, J.: Deep fruit detection in orchards. In: 2017 IEEE Int. Conference on Robotics and Automation (ICRA). pp. 3626–3633 (2017)
3. Femling, F., Olsson, A., Alonso-Fernandez, F.: Fruit and vegetable identification using machine learning for retail applications. In: 14th Int. Conf. on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 9–15. IEEE (2018)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
5. Hossain, M.S., Al-Hammadi, M., Muhammad, G.: Automatic fruit classification using deep learning for industrial applications. IEEE Trans. on Industrial Informatics **15**(2), 1027–1034 (2018)
6. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 7132–7141 (2018)
8. Katarzyna, R., Paweł, M.: A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. Applied Sciences **9**(19), 3971 (2019)
9. Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.: Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. Precision Agriculture **20**(6), 1107–1135 (2019)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Adv. in Neural Information Processing Systems 25, pp. 1097–1105 (2012)
11. Mureşan, H., Oltean, M.: Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica **10**(1), 26–42 (2018)
12. Rahnemoonfar, M., Sheppard, C.: Deep Count: Fruit counting based on deep simulated learning. Sensors **17**(4), 905 (2017)
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Inter. Journal of Computer Vision **115**(3), 211–252 (2015)
14. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C.: A fruit detection system using deep neural networks. Sensors **16**(8), 1222 (2016)
15. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
16. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Transactions on Medical Imaging **35**(5), 1285–1298 (2016)
17. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.: Efficient processing of deep neural networks: A tutorial and survey. Proc. of the IEEE **105**(12), 2295–2329 (2017)
18. Tan, L., Jiang, J.: Fundamentals of analog and digital signal processing. AuthorHouse (2007)