Detecting and Counting Pistachios based on Deep Learning

Mohammad Rahimzadeh
School of Computer Engineering
Iran University of Science and Technology, Iran
mh_rahimzadeh@elec.iust.ac.ir
ORCID: 0000-0002-8550-8967
Corresponding author

Abolfazl Attar
Department of Electrical Engineering
Sharif University of Technology, Iran
attar.abolfazl@ee.sharif.edu
ORCID: 0000-0001-6727-432X

August 30, 2020

Abstract

Pistachios are nutritious nuts that are sorted based on the shape of their shell into two categories: Open-mouth and Closed-mouth. The open-mouth pistachios are higher in price, value, and demand than the closed-mouth pistachios. Because of these differences, it is considerable for companies to precisely count the number of each kind. This paper aims to propose a new system for counting the different types of pistachios with computer vision. We have introduced and shared a new dataset of pistachios, including six videos with a total length of 167 seconds and 3927 labeled pistachios. At the first stage, we have trained RetinaNet, the deep fully convolutional object detector with three different backbones for detecting the pistachios in the video frames. In the second stage, we introduce our novel method for counting the open-mouth and closed-mouth pistachios in the videos. Pistachios that move and roll on the transportation line may appear as closed-mouth in some frames and open-mouth in other frames. Our work's main challenge is to count these two kinds of pistachios correctly and fast with this circumstance. Our algorithm performs very fast and achieves good counting results. The computed accuracy of our algorithm on six videos (9486 frames) is 94.75%.

Keywords: Deep learning; Convolutional Neural Network; Pistachio Counting; Multi-Object Counting; Object Detection; Motile-Object Counting;

1 Introduction

Nowadays, automation in the industry plays a significant role in increasing efficiency and saving resources. One of the industries that need more development in the field of automation than other industries is the agricultural industry and related fields. Proper packaging of agricultural products will increase profitability and reduce crop losses. On the other hand, crop quality categorization depends on human resources, which causes time-consuming and rising costs, and most importantly, does not have the necessary quality compared to machines.

It is one of the crops that need human resources to classify and count so that the quality of the crop can be evaluated in terms of its open or closed shell. Pistachios are mostly sorted based on the shape of their shell to open-mouth and closed-mouth, and these two kinds differ in the price and value.

Pistachios are used as nuts and in the food industry [39]. Pistachio kernels are rich in unsaturated fatty acids, fiber, carbohydrates, proteins and various vitamins that are very useful for the human diet [23, 14]. Adequate consumption of pistachio kernels reduces the risk of heart disease and has a good effect on blood pressure in people who do not have diabetes, and prevents some cancers[9, 14, 38]. Pistachio is one of the main agricultural products of Middle Eastern countries, especially Iran, [7]. The largest producers of this product in the world are Iran, USA and Turkey, respectively [7].

There are many types of pistachios, depending on the type and place of growth; they have different sizes, colors, and flavors [30]. Depending on the shape of the pistachio, it can be divided into three general categories: round, long, and jumbo [30]. Long pistachios have a narrower split than the other two, and the round and jumbo type have a much clearer split than the semi-closed one [30]. Fig 1 shows a summary of pistachios' different types.

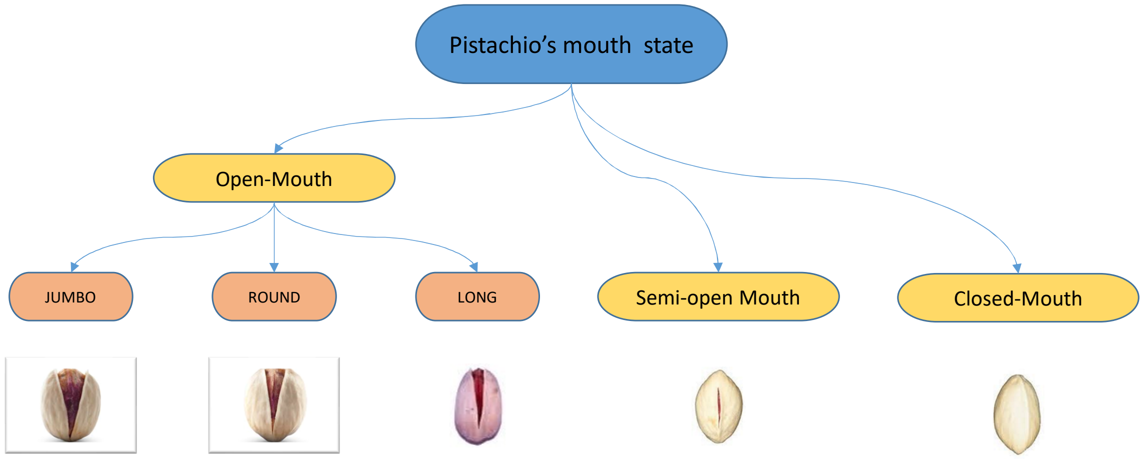

Figure 1: Pistachios Assortment

The average weight of each pistachio its shell is about 0.57 grams [15], which is about 1750 per kilogram. As a result, according to the statistics provided, counting them is a very time consuming and tedious task that can be easily done by artificial intelligence.

Detecting and counting the pistachios can be used for proper packaging and crop quality assessment. Another advantage of this can be the estimation of the amount of the crops in the coming years and the breeding of pistachio trees to increase the quality of the crop. As there is a significant difference in price and demand between the open-mouth and closed-mouth pistachios, the factories related to pistachio production or packaging, need to know precisely how much of these two kinds exist in every package. Counting these two kinds of pistachios can also help separate them to increase the quality of the exporting packages. Counting the pistachios by human resources is very time consuming and practically uneconomical so that machine vision can play a significant role in this regard.

Another application of these procedures is that closed pistachios are used by the mechanical opening method [3] to be returned to the consumption cycle. In this method, it is first necessary to identify the closed pistachios, which results in reduced losses and increased crop yields [3].

One of the new methods for detecting, counting, and classifying the pistachios is machine vision. In recent years, machine vision has been used for many tasks to automate and replace machines with humans, which have yielded excellent results [32]. These applications exist in the fields of medicine [18], medical image diagnosis [26, 28], self-driving [10], security [2], and agriculture [29, 22, 24], and so on.

One of the principles of using robots and remote control and sense is the use of machine vision. Therefore, improving the accuracy and precision of the system is one of the essential principles. In machine vision, various methods such as thermal cameras, sensors, microscopes, and common cameras have been used for imaging space around it. However, the main issue in machine vision is the choice of the data analysis method.

Currently, one of the most attractive and accurate methods of machine vision is deep learning, which has been created a revolution in artificial intelligence [17]. One of the most important advantages of deep convolutional neural network is that it is comprehensive and flexible in recognizing different objects. [16].

Using the deep convolutional neural networks, we can identify and count the pistachios. Depending on the appearance of the pistachio, the angle of the camera or robot also plays a vital role in correctly identifying the open and closed pistachios. Another critical challenge is to count the open-mouth and closed-mouth pistachios correctly because the open-mouth pistachios can show themselves as closed-mouth pistachio when moving and spinning on the transportation line.

In this paper, at first, we will propose our dataset, which we call Pesteh-Set. At the next stage, we will describe the detection phase. We have used RetinaNet [20] as the object detector for detecting the Pistachios in the video frames. We have separated the dataset into five-folds and allocated 20 percent of the dataset for testing and the rest for the training. After the detection phase, we present the method we used for counting the open-mouth and closed-mouth

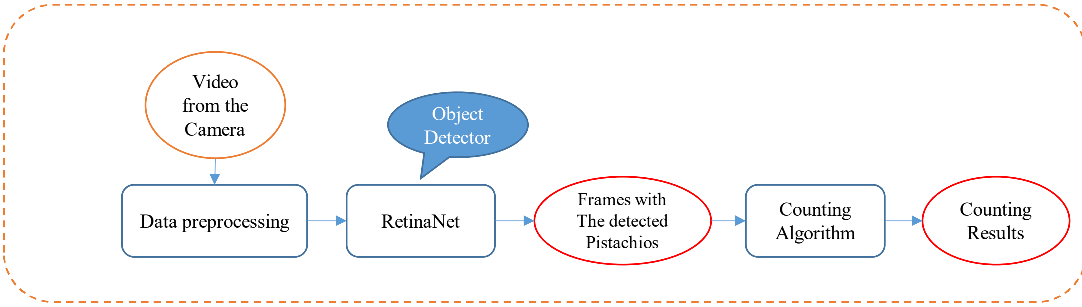

Figure 2: General Schematic of our proposed method

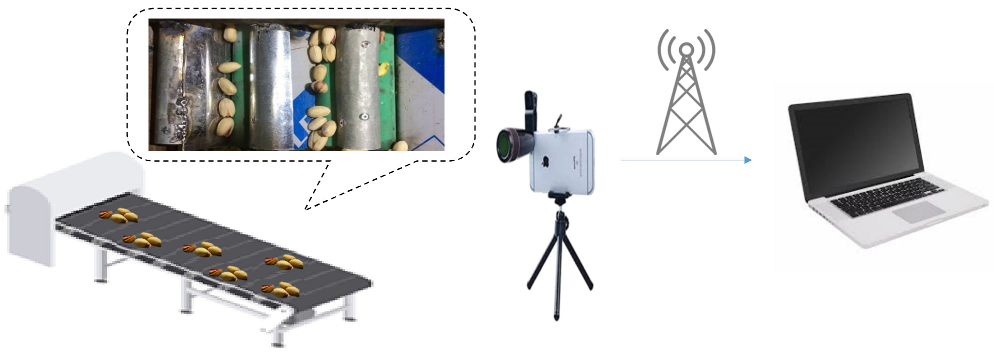

Figure 3: The General View of how Pesteh-Set was recoreded and our proposed way for counting the pistachios

pistachios. This algorithm runs very fast with high accuracy. The general schematic of our work is presented in fig 2.

One of the closest studies in our research is fruit detection and counting [11]. This is done using a variety of tools such as a B/W camera [37], a color camera [5], a thermal camera [35] and a spectral camera [31]. Due to the type of data we have, color is one of the most important features, B/W camera is not suitable [37]. On the other hand, because the spectral camera has a time delay, this method cannot be suitable [31]. Due to its sensitivity to size and lack of detection of split pistachios, the thermal camera is not suitable for analyzing our data [35]. As a result, the color camera is more suitable than other tools.Another advantage of using color cameras is its abundance, especially in mobile phones, which can be used for remote monitoring and control.

There are other methods such as sensors that can be used for counting, but since one of the challenges ahead is split and non-split pistachios counting, however, this method is not suitable for our data [8].

One of the most widely used classifications and detection methods used is the K-means clustering method, which is performed unsurprisingly. In [36], K-means clustering detects green apples using thermal and color cameras.

One of the most basic methods of Supervised classification is the Bayesian classifier, which has been used in [34] to identify oranges and has yielded relatively good results from previous research. Other used methods include KNN clustering [21].

Artificial neural networks have a special place in machine vision and object detection. In the meantime, deep convolutional neural networks have shown excellent results for images and videos, too. Therefore, image detection systems move to increase the accuracy and quality of deep learning.

In [24], they classify different fruits using an innovative deep neural network. In [29], a deep neural network was developed to detect and count the number of tomatoes per plant. In this study, due to the lack of enough data, the data were generated by simulating a green and brown environment in which the tomatoes were simulated with red circles, which this can take the results away from the real world.

Table 1: The distribution of Pistachios in Pesteh-Set

	Number of Open-mouth Pistachios	Number of Closed-mouth Pistachios	Number of All the Pistachios
Video 1	50	20	70
Video 2	60	20	80
Video 3	70	20	90
Video 4	90	20	110
Video 5	100	20	120
Video 6	39	52	91
All of the Videos	409	152	561
All the 423 labeled images	1993	1934	3927

In [22], the environment is photographed using a monocular camera, and visible fruits are detected and tracked on the tree. This detection is made by training a fully Convolutional Network, then using image processing methods to track the fruits, and then counting the fruits.

[27] is one of the papers that has done very well in detecting and tracking motile objects. This paper also introduced a method for improving motile-objects detection. In this article, by using RetinaNet [20], and the introduced method, they have detected motile sperms in the video frames. Finally, by using the modified CSR-DCF, they have tracked and analyzed the sperms attributes, such as their number and motility characteristics.

The rest of the paper is organized as follows: In section 2, we talk about our dataset and detection and counting phases. In section 3, the results of our work are presented. In section 4, we discuss the obtained results, and in section 5, the paper is concluded.

2 Materials and methods

2.1 Pesteh-Set

Pistachio is known as Pesteh in Iran, that is why we called our dataset Pesteh-Set. Pesteh-Set [1] is made of two parts. The first part includes 423 images with ground truth. We sorted the pistachios into two classes: Open-mouth and closed-mouth. The ground truth of the images consists of the bounding boxes of the two classes of pistachios in the images. There are between 1 to 27 pistachios in each image, and 3927 pistachios totally. The second part includes six videos (9486 frames) that were used for the counting phase. These six videos include 561 motile pistachios and more than 350,000 single pistachios (sum of pistachios in each frame).

The videos of the dataset have been recorded by a cell-phone camera with 1920 $\times$ 1080 pixels resolution, five of these videos are recorded with 60 frames per second(fps) frame rate, and one other is recorded with 30 fps frame rate. The cell-phone was perched on the wall above the line that was transporting the pistachios. This line was designed somehow that the pistachios could roll on it. The reason the pistachios rolling is so important is that the open-mouth pistachios could appear on their backside where they look like closed-mouth pistachios, but the rolling cause them to show their open-mouth side when rolling. Fig. 3 presents a view of how the dataset was recorded, and also the general schematic of our proposed method for remote counting the pistachios.

We have selected some frames of the videos and labeled them with a self-developed program using OpenCV library [25] on python language. The images of the dataset were resized to 1070 $\times$ 600 pixels to save computing costs. The pistachios are categorized into two classes: open-mouth pistachios and closed-mouth pistachios. Some of the images of this dataset are presented on fig. 4 The self-developed program for labeling the images along all the data has been shared so other researchers could use them to make the Pistachio-Dataset larger. Table1 presnets the details of Pesteh-Set.

In table 1, the reason that the number of pistachios in the videos is less than the images is that the number of pistachios in the videos denotes the number of mobile pistachios. It means from where one pistachio enters the video in a frame

[1] <This dataset is shared in https://github.com/mr7495/Pesteh-Set>

Figure 4: Some of the images in Pesteh-Set

until it exits the video in the later frames, it would be counted as one pistachio. In the images, we counted the number of pistachios in each image. It is noteworthy that we selected non-consecutive frames of different videos, so there would not be a similarity between them. Besides, we tried to choose the frames somehow that we have an almost equal number of each class, and our dataset become balanced for training.

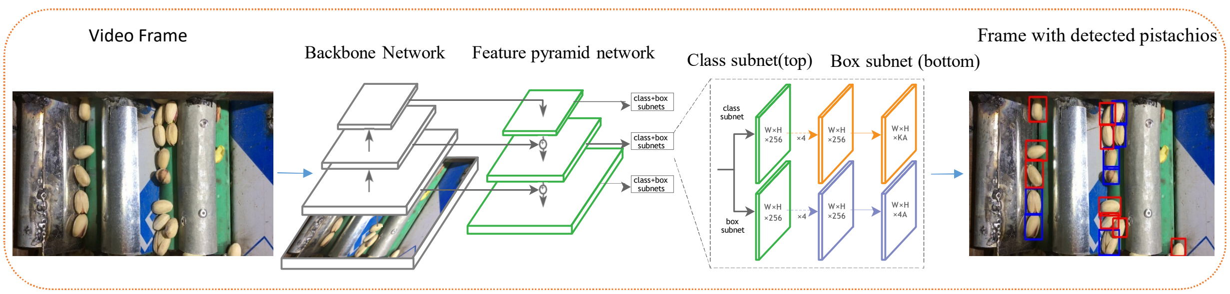

Figure 5: RetinaNet Architecture

2.2 Detection

2.2.1 RetinaNet

RetinaNet [20] is a deep fully convolutional neural network that is utilized for object detection. The architecture of RetinaNet is depicted in fig 5. RetinanNet is made of three main parts. The first part, which is the feature extractor, is build up from the main feature extractor that is called the backbone and the feature pyramid network(FPN) [19]. Famous feature extractor convolutional networks like ResNet [12], DenseNet [13], and VGG [33] are mostly used as the backbone of the RetinaNet. The FPN takes the multi-dimensional features that are extracted from the backbone network as the input to build a multi-scale feature pyramid from the input image [19]. The usage of the FPN on top of the backbone considerably improves the object detection accuracy.

The second part of the RetinaNet is the Classifier, which has the role in predicting the possibility of the presence of each of the classes at each spatial location for each of the anchor boxes. The second part is the box regression that regresses each of the anchor boxes to the nearest ground truth object boxes [20]. Another novelty presented in the RetinaNet is using the focal loss [20] as the loss function. The focal loss adds a modulating factor to the cross-entropy loss function, and by doing so, it focuses on the hard examples while training, and as the loss of hard examples is higher than the easy examples, it causes to improve the learning process and accuracy.

2.2.2 Training

We have separated the Pesteh-Set into five folds for training, which in each fold, 20 percent of the dataset was allocated for testing, and the rest for training. The images of the dataset were preprocessed and then resized to 1070×600 pixels.

We used RetinaNet [20], as the object detector. We trained and validated RetinaNet on 3 different backbones: ResNet50 [12], ResNet152 [12], and VGG16 [33]. Transfer learning from the ImageNet [6] pre-trained weights was utilized at the beginning of the training to speed up the network convergence. We also used data augmentation methods to improve the learning efficiency and stop the network from overfitting. The applied training parameters are listed in the table 2 and the details of each fold are present in table 3.

Training Parameters	Value
Learning Rate	1e-5 (With automatic reductionbased on Loss value)
Batch Size	1
Optimizer	Adam
Loss Function	Focal Loss for the classification subnet Smooth L1 for the regression subnet
Steps	1017
Horizontal/Vertical flipping	Yes (50%)
Translation Range	-0.1 - 0.1
Rotation Range	0 - 360 degree
Shear Range	-0.1 - 0.1
Scaling Range	-0.1 - 0.1

Table 2: This table shows all the parameters and methods we used in training

2.3 Counting

The second and main phase of our work was counting the number of open-mouth and closed-mouth pistachios in the videos. To do so, first, we used a frame generator to extract the frames of the video, then we fed the frames to the object detector, and finally, we had a list of bounding boxes for each frame.

There were several challenges in this phase. The first challenge was that we wanted to develop a method that could be performed very fast on the CPU. Some of the other ideas may need a GPU; otherwise, the process would become extremely time-consuming. However, our method works very much fast on CPU, even faster than the methods that need to be executed on GPU.

The second challenge was that some of the open-mouth pistachios could show themselves as closed-mouth pistachio in some consecutive frames and then reveal their open part only a few frames. Moreover, some open-mouth pistachios that are rolling on the transportation line could show their open part several times and then be appeared like closed-mouth pistachios like fig 6. We had to develop our algorithm somehow to prevent failing because of these challenges.

Table 3: This Table presents the details of our train and validation sets in each fold

Fold Number	Training Images	Validation Images	Open-mouth Pistachios in Training Set	Closed-mouth Pistachios in Training Set	Open-mouth Pistachios in Validation Set	Closed-mouth Pistachios in Validation Set
Fold 1	339	84	1600	1550	393	384
Fold 2	339	84	1610	1572	383	362
Fold 3	339	84	1553	1506	440	428
Fold 4	339	84	1641	1575	352	359
Fold 5	336	87	1568	1533	425	401

Another challenge is to develop the counting method in a way to prevent failure because of false detections or not-detected pistachios that may be affected by the pistachios occlusion.

For counting the pistachios, we first generate the frames from the taken video and then use the trained network for getting the bounding boxes of the pistachios in the frames. After this process, we would have a list of bounding boxes from the consecutive frames of the video. Then the algorithm has to deal with a list of data, and that is why the counting process performs very fast.

Two thresholds have been designed to improve the counting accuracy: the initial threshold and the end threshold. The initial threshold is set to detect the newly inserted pistachios, and the end threshold is to reject adding the pistachios to the track list. These two methods are explained in the next paragraphs. As the height of our image is 600 pixels, we set the end threshold equal to 500. We call the area of the image with height less than the initial threshold, the entering area, and the area with a height higher than the end threshold, the exiting area.

The algorithm first runs a function to set the initial threshold. In this function, the algorithm begins to assign the pistachios between each of two consecutive frames based on their distance without considering the class of pistachios. The minimum acceptable distance to assign the pistachios has been set to 20 pixels. The number of 20 pixels is taken out of our experience to prevent false assignments. After the assigning, the function adds the not-assigned pistachios that the height of the mid-point of their bounding box be less than 200 (the height of the images is 600 pixels), to a list. These added pistachios are candidates as new inputs. After adding all the eligible pistachios, the function measures the average of the list, and it would be set as the initial threshold. This process performs to measure the area that most of the pistachios will enter the frame. The pistachios in different videos can enter the video frames differently and also may have various speeds, so this function will set the initial threshold wisely to improve the counting.

In the next level, the algorithm uses the assigned pistachios of each of the two consecutive frames, that were computed in the last step. Our counter algorithm role is not only to count the number of all pistachios but also to count the number of open-mouth and closed-mouth pistachios. The algorithm uses the initial threshold and the end threshold, which are computed in the last step. Toward solving the main challenge, which is that many of the pistachios may show their open part in some frames and the closed parts in the other frames, we have to track them by assigning them from frame to frame. We decided to track them from when they enter the entering area until they enter the exiting area. By doing this, we can know if one pistachio is open-mouth or closed-mouth, and it also prevents the algorithm from counting extra open-mouth pistachios.

The algorithm analyzes each of the assigned pistachios in the two consecutive frames, for all the frames. If the pistachio in the last frame was in the existing area or this pistachio in the current frame is in the exiting area, this pistachio would also be rejected to be added to the track list. Otherwise, the pistachio in the current frame would be assigned to the track list that the assigned pistachio in the previous frame belongs to that. After adding all the assigned pistachio in the current frame to the track lists that they belong to, the algorithm investigates the pistachios in the current frame that have not been assigned to any other pistachios. If these pistachios are located in the exiting area, they would be rejected to be added to the track list. If they be in the entering area and the number of all the pistachios in the current frame be greater than the number of pistachios in the last frame, these pistachios would be considered as new inputs and would start their own track list. The reason the pistachios in the current frame must be higher than the pistachios in the last frame is that in most cases if new pistachio enters the frame, the number of all frames should increase but you may think that this situation may not always happen. It is true, but it also equalizes the conditions that some other pistachios be in the entering area in the current frame, but not be detected in the last frame, so they cannot be assigned, and that pistachio in the current frame will be assumed as a new input. The unassigned pistachios in the current frame that can not be chosen as new inputs would be entered into the Lost-Pistachios list.

The Lost-Pistachios list is created for assigning the pistachios that could not be assigned to the last frame pistachios (maybe because of that the pistachios in the last frame not be detected), to the pistachios in the 2 to 6 previous frames. In the last stage, the counting algorithm tries to assign the pistachios in the Lost-Pistachios list to the pistachios in the 2 to 6 previous frames that were not assigned to any other pistachios also. If the assignment is successful, the newly assigned pistachio will be added to the track list, and if not, it will be rejected.

Finally, after repeating this procedure for all the consecutive frames, we would have a list of tracked pistachios. If in a track there be an open-mouth pistachio, the whole track will be considered as open-mouth, therefor so we could count the open-mouth and closed -mouth pistachios. The flow chart of the proposed counting algorithm is presented in fig 8.

3 Results

3.1 Detection Results

We trained Retinanet with three different backbones: ResNet50 , ResNet152 and VGG16 based on the explained parameters in table 3 for 50 epochs. Data augmentation methods like rotation, translation, shearing, horizontal and vertical flipping, and rescaling were also applied to improve the training and prevent overfitting.

The system we used in this paper was provided by Google Colaboratory Notebooks, which allocated a Tesla P100 GPU, 2.00GHz Intel Xeon CPU, and 25GB RAM on Linux to us. For utilizing RetinaNet we used the written codes by Fizyr which implemented RetinaNet with Keras library [4] on Tensorflow backend [1]. The metrics we used for evaluating RetinaNet in the detection phase are: Recall, Precision, F1 score, Accuracy, Average Precision(AP)and Mean Average Precision(map).

AP is defined as:

$$AP = \frac{\sum_{i=1}^{D} \{Precision(i) \times Recall(i)\}}{Number\ of\ annotations} \tag{1}$$

In Eq. 1, D is the number of detected pistachios that sorted by scores. Average Precision will be calculated for each class separately. The map metric is the mean of Average Precision between classes.

Fig 7 presents some of the images with the detected pistachios.

To evaluate the detection results, we considered the detected boxes with Intersection over Union (IOU) more than 0.5 as true positives and the others as false positives. We have reported the detection results in table The detection results of RetinaNet on the three backbones are presented in table 5 and 6.

3.2 Counting Results

Six different videos with 167 seconds length and 9486 frames were selected for evaluating the counting algorithm. We tested our counting algorithm based on the detections gathered from the trained networks on different backbones. The results and the overall accuracy for all the videos are present in table 7.

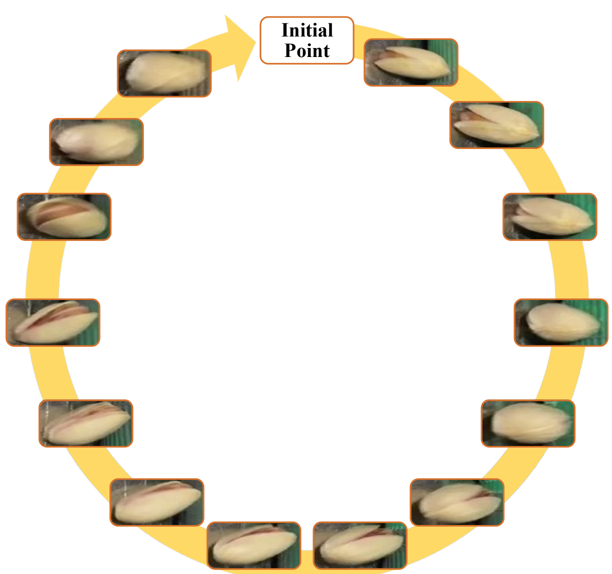

Figure 6: In this figure you can observe that a pistachio can be presented as open-mouth and closed-mouth several times while moving.

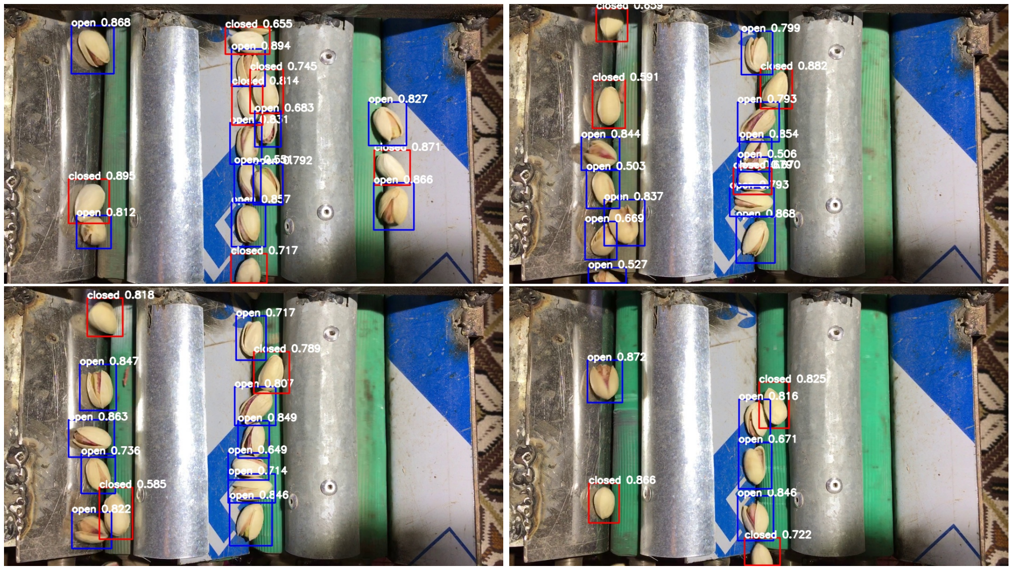

Figure 7: These images are the output of RetinaNet. The red boxes are the closed-mouth pistachios, and the blue boxes belong to the open-mouth pistachios. The number beside the open or closed is the detection score value.

Video	Number of Frames	Time (S)
Video1	984	3.00
Video2	1665	3.39
Video3	1833	5.11
Video4	2227	5.75
Video5	2171	4.19
Video6	606	0.83

Table 4: In this table the time, our counter algorithm takes to run (after getting the detections from RetinaNet) is reported for each of the tested videos.

Table 4 expresses the running time for the counter algorithm. We have used the accuracy metric for evaluating the tracking algorithm which is defined as:

$$Accuracy = \frac{TP}{TP + FN + FP} \tag{2}$$

In equation 2, TP is the number of the correct-counted pistachios, FN is the number of not-counted pistachios, and FP is the number of extra miscounted pistachios.

4 Discussion

Based on the table 6, in fold 1 and 4, ResNet152 performs better, but in other folds, ResNet50 achieves better results. One of the reasons that the reported metrics are not very high is because the open-mouth and the closed-mouth pistachios could look like each other in many cases like fig 9, and it would be hard to distinguish them even with human eyes. It can be hoped that in the future works, researchers use our shared videos and the developed program for labeling the images, to make more data with ground truth and improve the training accuracy.

The point we can see is that although the detection accuracy is not very high, the counting results based on table 7 are promising. This shows our proposed counter algorithm's robustness, which has been evaluated on six different videos containing 9486 frames with 561 moving pistachios and more than 350,000 single pistachios (sum of pistachios in each frame). The counting procedure based on table 4 has been performed very fast. Because of this high speed and good accuracy, this algorithm is capable of being utilized in factories and industries related to pistachios.

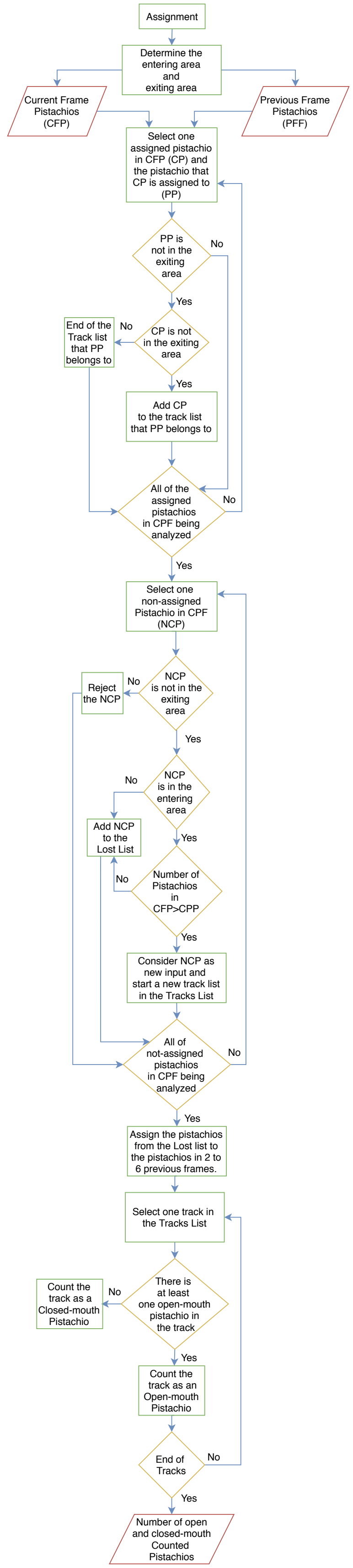

Figure 8: The flow chart of the proposed counting algorithm

		Class closed-pistachio			Class open-pistachio		
		AP	F1 score	Recall	AP	F1 score	Recall
Fold1	ResNet50	0.9072	0.9128	0.9270	0.9467	0.9325	0.9669
	ResNet152	0.9037	0.9238	0.9166	0.9686	0.9302	0.9847
	VGG16	0.8821	0.9067	0.8984	0.976	0.9397	0.9923
Fold2	ResNet50	0.9135	0.9213	0.9226	0.9172	0.9359	0.9347
	ResNet152	0.9010	0.9105	0.9143	0.9151	0.9286	0.9347
	VGG16	0.8841	0.8856	0.9198	0.9157	0.8988	0.9399
Fold3	ResNet50	0.9402	0.9232	0.9556	0.8832	0.9130	0.9068
	ResNet152	0.9096	0.9161	0.9322	0.8981	0.9171	0.9181
	VGG16	0.8919	0.9138	0.9042	0.8922	0.9032	0.9227
Fold4	ResNet50	0.9183	0.9115	0.9331	0.9301	0.9258	0.9403
	ResNet152	0.9347	0.9243	0.9526	0.9449	0.9318	0.9517
	VGG16	0.9186	0.9110	0.9415	0.9303	0.9194	0.9403
Fold5	ResNet50	0.9122	0.9106	0.9276	0.9179	0.9175	0.9294
	ResNet152	0.8830	0.8996	0.9052	0.9100	0.9160	0.9247
	VGG16	0.8922	0.8907	0.9152	0.8979	0.9153	0.9035

Table 5: The detection results for each class of RetinaNet on different backbones for each fold are reported in this table.

	Backbone Network	TP	FP	FN	Recall	Precision	F1 score	map	Accuracy
Fold1	ResNet50	736	82	41	0.9472	0.8997	0.9228	0.9270	0.8568
	ResNet152	739	78	38	**0.9510**	**0.9045**	**0.9272**	**0.9361**	**0.8643**
	VGG16	735	79	42	0.9459	0.9029	0.9239	0.9291	0.8586
Fold2	ResNet50	692	53	53	0.9288	**0.9288**	**0.9288**	**0.9153**	**0.8671**
	ResNet152	689	64	56	0.9248	0.9150	0.9198	0.90812	0.8516
	VGG16	693	115	52	**0.9302**	0.8576	0.8924	0.8999	0.8058
Fold3	ResNet50	808	84	60	**0.9308**	0.9058	**0.9181**	**0.9117**	**0.8487**
	ResNet152	803	81	65	0.9251	**0.9083**	0.9166	0.9038	0.8461
	VGG16	793	85	75	0.9135	0.9031	0.9083	0.8921	0.8321
Fold4	ResNet50	666	73	45	0.9367	0.9012	0.9186	0.9242	0.8494
	ResNet152	677	71	34	**0.9521**	**0.9050**	**0.9280**	**0.9398**	**0.8657**
	VGG16	669	82	42	0.9409	0.8908	0.9151	0.9244	0.8436
Fold5	ResNet50	767	85	59	**0.9285**	0.9002	**0.9141**	**0.9151**	**0.8419**
	ResNet152	756	83	70	0.9152	**0.9010**	0.9081	0.8965	0.8316
	VGG16	751	86	75	0.9092	0.8972	0.9031	0.8950	0.8234
Average	ResNet50	**733.8**	**75.4**	**51.6**	**0.9344**	**0.9071**	**0.9205**	**0.9187**	**0.8528**
	ResNet152	732.8	**75.4**	52.6	0.9336	0.9068	0.9199	0.9169	0.8519
	VGG16	728.2	89.4	57.2	0.9279	0.8903	0.9086	0.9123	0.8332

Table 6: This table contains the RetinaNet evaluation data for all the classes.

Backbone Network	Ground-Truth Open-Mouth Pistachios	Ground-Truth Closed-Mouth Pistachios	Correctly Counted Open-Mouth Pistachios	Correctly Counted Closed-Mouth Pistachios	Extra Counted	Accuracy
ResNet152	409	152	397	145	11	**0.9475**
ResNet50	409	152	386	152	24	0.9196
VGG16	409	152	395	149	37	0.9096

Table 7: Counting results for all the 6 test videos. The detections are taken from the trained networks in the first fold. Extra counted means the sum of miscounted open-mouth and closed-mouth pistachios.

Figure 9: Some of the examples in our dataset which are hard to classify

5 Conclusion

Pistachio is a nutritious nut that originated from central Asia and the middle east, and some countries are famous for exporting it. In this paper, we have proposed a dataset and some novel methods that can be used to design a remote AI system to detect and count the number of open-mouth and closed-mouth pistachios in the production line of the factories that are related to pistachio production or packing. These methods can be run by a regular camera that records from the production line and a qualified system that can be used for running the algorithms.

We have introduced a new dataset that we called Pesteh-Set. Pesteh-Set is made of 6 videos (9486 frames) and 3927 labeled pistachios of two classes: open-mouth and closed-mouth pistachios. Pistachios are motile objects and usually spin on the transporting line, which makes our work's main challenge. This challenge is that it often happens that in some frames of the video, some open-mouth pistachios place on their backside and look like closed-mouth pistachios. Due to this challenge, we had to develop our methods somehow to prevent false counting. Another challenge was to design a counting system that could be performed fast. For counting the pistachios first, we used RetinaNet, the deep fully convolutional object detector, to detect the pistachios in the video frames. We trained RetinaNet on three different backbones: ResNet50, ResNet152, and VGG16 in five folds. The average f1 score for RetinaNet on ResNet50 network was 92.05%. The detection accuracy can be increased in future works. One way is that researchers can use our developed program and the dataset to generate more labeled images. They can label all the frames of some videos and use this method [27] that was proposed to improve the motile-objects detection. Pistachios are motile objects, so by using this method, the detection accuracy should be improved.

After getting the detections, we implemented our proposed counter method to count the open-mouth and closed-mouth pistachios. Our counter method performed fast, with no need for GPU (other than the object detection part), and achieved good results. This counting algorithm was tested on six different videos containing 9486 frames with 561 moving pistachios and more than 350,000 single pistachios (sum of pistachios in each frame). This algorithm obtained 94.75% accuracy when the detections were taken from RetinaNet on ResNet152 backbone. This work can be extended to detect and count more than two classes of pistachios, e.g., .semi-open mouth pistachios can also be added to the classes.

6 Data Availability

In this GitHub profile (`https://github.com/mr7495/Pesteh-Set`), we have shared our dataset and all the codes that were used for preparing and labeling the dataset.

7 Code Availability

In this GitHub profile (`https://github.com/mr7495/Pistachio-Counting`), we made the trained neural networks, the counting algorithm and all the codes that were used for training and validating the networks, public for researchers use.

Acknowledgment

We wish to thank Mr.Navid Akhundi, who recorded our dataset videos and shared them with us, and Fizyr, who implemented RetinaNet with Keras on GitHub.

We also thank Colab server for providing free and powerful GPU and Google Drive for providing space for data hosting.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett. A survey of deep learning methods for cyber security. *Information*, 10(4):122, 2019.

[3] C. D. Burlock, G. E. Lemmons, and D. W. Williams. Apparatus for splitting closed shell pistachio nuts, Mar. 5 1991. US Patent 4,996,917.

[4] F. Chollet and Others. keras, 2015.

[5] O. Cohen, R. Linker, and A. Naor. Estimation of the number of apples in color images recorded in orchards. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 630–642. Springer, 2010.

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

[7] 2018 (accessed May 2 2020).

[8] J. Feng, G. Liu, S. Wang, L. Zeng, and W. Ren. A novel 3d laser vision system for robotic apple harvesting. In *2012 Dallas, Texas, July 29-August 1, 2012*, page 1. American Society of Agricultural and Biological Engineers, 2012.

[9] Food, D. Administration, et al. Qualified health claims: Letter of enforcement discretion–nuts and coronary heart disease. docket no. 02p-0505. *Food and Drug Administration, Washington, DC*, 2003.

[10] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[11] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis. Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, 116:8–19, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[14] M. Kashaninejad, A. Mortazavi, A. Safekordi, and L. Tabil. Some physical properties of pistachio (pistacia vera l.) nut and its kernel. *Journal of Food Engineering*, 72(1):30–38, 2006.

[15] P. J. Kiger. Why pistachios are sold in their shells - unlike most nuts, Mar 2017 (accessed May 2 2020).

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

[17] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[18] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim. Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570–584, 2017.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[21] R. Linker, O. Cohen, and A. Naor. Determination of the number of green apples in rgb images recorded in orchards. *Computers and Electronics in Agriculture*, 81:45–57, 2012.

[22] X. Liu, S. W. Chen, S. Aditya, N. Sivakumar, S. Dcunha, C. Qu, C. J. Taylor, J. Das, and V. Kumar. Robust fruit counting: Combining deep learning, tracking, and structure from motion. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1045–1052. IEEE, 2018.

[23] M. Maskan and Ş. Karataş. Fatty acid oxidation of pistachio nuts stored under various atmospheric conditions and different temperatures. *Journal of the Science of Food and Agriculture*, 77(3):334–340, 1998.

[24] H. Mureşan and M. Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 10(1):26–42, 2018.

[25] OpenCV. Open source computer vision library, 2015.

[26] M. Rahimzadeh and A. Attar. A modified deep convolutional neural network for detecting covid-19 and pneumonia from chest x-ray images based on the concatenation of xception and resnet50v2. *Informatics in Medicine Unlocked*, page 100360, 2020.

[27] M. Rahimzadeh, A. Attar, et al. Sperm detection and tracking in phase-contrast microscopy image sequences using deep learning and modified csr-dcf. *arXiv preprint arXiv:2002.04034*, 2020.

[28] M. Rahimzadeh, A. Attar, and S. M. Sakhaei. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. *medRxiv*, 2020.

[29] M. Rahnemoonfar and C. Sheppard. Deep count: fruit counting based on deep simulated learning. *Sensors*, 17(4):905, 2017.

[30] Different types of iranian pistachios and products of pistachios, Jan 2020 (accessed May 2 2020).

[31] O. Safren, V. Alchanatis, V. Ostrovsky, and O. Levi. Detection of green apples in hyperspectral images of apple-tree foliage using machine vision. *Transactions of the ASABE*, 50(6):2303–2313, 2007.

[32] A. Shrestha and A. Mahmood. Review of deep learning algorithms and architectures. *IEEE Access*, 7:53040–53065, 2019.

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

[34] D. C. Slaughter and R. C. Harrell. Discriminating fruit for robotic harvest using color in natural outdoor scenes. *Transactions of the ASAE*, 32(2):757–0763, 1989.

[35] D. Stajnko, M. Lakota, and M. Hočevar. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. *Computers and Electronics in Agriculture*, 42(1):31–42, 2004.

[36] J. P. Wachs, H. I. Stern, T. Burks, and V. Alchanatis. Low and high-level visual feature-based apple detection from multi-modal images. *Precision Agriculture*, 11(6):717–735, 2010.

[37] D. Whittaker, G. Miles, O. Mitchell, and L. Gaultney. Fruit location in a partially occluded image. *Transactions of the ASAE*, 30(3):591–0596, 1987.

[38] Pistachio, Apr 2020 (accessed May 2 2020).

[39] J. G. Woodruff et al. *Tree nuts: production, processing, products*. Number Ed. 2. AVI Publishing Co. Inc., 1979.