DeepNut: A Finding Nemo Task

1st Deepan Chakravarthi Padmanabhan *Masters. Autonomous systems*
Hochschule Bonn-Rhein-Sieg
Bonn, Germany
deepan.padmanabhan@smail.inf.h-brs.de

Abstract

Nut detection is an imperative aspect given the substantial health benefits of nuts and the increasing growth of the edible nut industry. To characterize and analyze a nut, the process of detecting and classifying the nuts is an imperative task. The pivotal aspect of this project is to detect three classes of nuts, namely, peanut, walnut, and hazelnut. The proposed solution is robust to variations in the illumination conditions, viewpoints, frame rates, backgrounds, and the number of distractors in the video input. This particular work aims to deploy a deep learning based object detection methods to detect nuts. In addition, the project utilizes classical methods such as Contrast Limited Adaptive Histogram Equalization (CLAHE) for preprocessing and frame differencing for stable frame extraction from the video input. The nuts are detected within a confined rectangular area of a stable frame extracted from video input. MobileNet-SSD is the deep learning model trained. The model achieves a mean Average Precision (mAP) of 88.12% on the validation set generated from the CV 2019 dataset with an average inference time of 1.72 seconds. The developed DeepNut detector is evaluated using a localization efficiency metric and mean F-score of all the nut classes for localization and classification tasks, respectively. DeepNut detector localizes the nuts with an efficiency of 83.33% and classifies the nuts correctly with a mean F-score of 0.921.

Index Terms

Deep learning, Single-shot object detector (SSD), MobileNet-SSD, mean Average Precision (mAP), frame differencing, Contrast Limited Adaptive Histogram Equalization (CLAHE).

I. Introduction

Edible nuts provide various health benefits, namely, cardiovascular diseases, inflammation, hyperglycemia, and diabetes [10]. The edible nut market is expected to grow at a Compound Annual Growth Rate (CAGR) of 3.5% between 2018 and 2026 [21]. In addition, nuts are prone to fungal and bacterial infections. For instance, a fungal mold-infested walnut spoils the batch and creates aflatoxin - a potent carcinogen [34]. These factors raise the need for edible nut quality assessment. The United States Food and Drug Administration (FDA) perform visual and organoleptic examination involving huge human efforts for testing nuts [22]. Therefore, the need to automatically localize and classify nuts is significant.

Object detection is a Computer Vision (CV) task for localization and classification of different classes of objects in digital images. Object detection models answer the question *what objects are where?* using traditional or deep learning approach. However, the review studies [25] [26] comparing deep learning and traditional approaches conclude both methods have their own advantages. In addition, the papers conclude the approach to be selected for a particular task depends on the problem at hand [25] [26]. This paper utilizes a hybrid approach to localize and classify three types of nuts, namely, walnut, peanut, and hazelnut in a dataset involving a variety of videos.

The study utilizes frame differencing, Contrast Limited Adaptive Histogram Equalization (CLAHE) followed by deep learning based object detection to detect the nuts of interest. Recent advances in the deep learning architectures, mainly Convolutional Neural Networks (CNN) and the advent of parallel computing through Graphical Processing Units (GPU) has taken object detection a step forward. Deep learning-based object detection methods have proved to perform way better than traditional detection methods that use handcrafted features such as SIFT [6] and HoG [7] [2]. The ability of CNN architectures such as VGG [12], Inception [13], DenseNet [15], ResNet [16] to represent the high-level features of the image is one of the reasons for the remarkable performance of the state-of-the-art object detectors. These

CNN architectures form the backbone of most state-of-the-art object detectors. Along with the CNN architectures, the region proposal network-based object detectors like Faster RCNN [8], R-FCN [17] and Mask R-CNN [18] achieved extraordinary detection results. Moreover, object detectors like YOLO [9], SSD [19] and YOLOv2 [20] have proved to achieve remarkable results despite taking fewer computations [2].

Object detection using deep learning demands high computational power, and hence model compression techniques provide a key contribution in storing the models on the memory efficiently. Techniques like squeezeNet [27] and MobileNets [28] offer high model compressions while retaining the accuracy levels. MobileNet-SSD provides relatively faster inference, and performance is relatively equivalent to other state-of-the-art models [29]. Therefore, this project utilizes MobileNet-SSD [29] for efficiently detecting nuts in the challenging environment provided.

II. Problem description

This paper addresses the problem of localizing and classifying three different classes of nuts - peanut, walnut, and hazelnut in a stable frame of video data. Each video includes the actions of throwing the nuts along with distractors into a rectangular box, allowing the objects to settle and picking up all the objects. The stable frame is the frame where all the objects in the video are static without any motion.

The task is challenging because of the following reasons:

1) Each video in the dataset varies in illumination, viewpoint, frame rate, background, number of nuts, and number of distractors. The distractor objects include pen, cork, dice, bottle caps, and nuts (other than a walnut, peanut, and hazelnut). The figure 1 shows the distractor in the video. The figures 1 and 2 illustrate the differences in lighting conditions across the dataset.
2) Intra-class and inter-class variations among the objects of interest. The former is due to substantial variations within a single type of object, shown in figure 1, and in the latter situation, different types of objects look similar, shown in figure 2. For instance, the red dice in an under-exposed video frame contribute to the inter-class challenge due to their similarity with small hazelnuts, shown in figure 2.

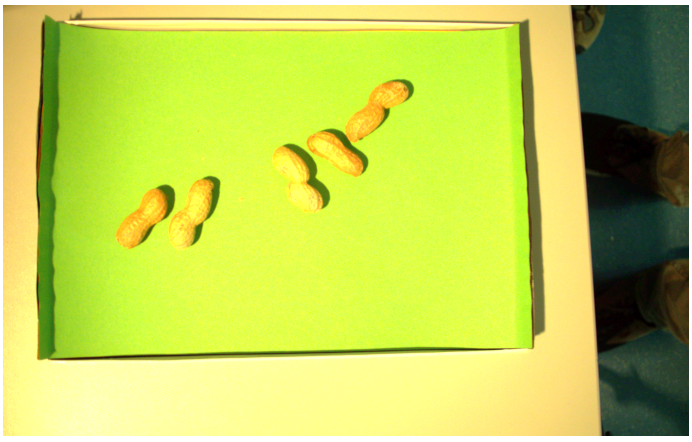


Fig. 1. Illustration of intra-class variation (Stable frame of video CV19_video_30.avi). The different shapes of peanut in a single frame illustrate the intra-class variations challenge.

III. Related work

Kuo-Yi-Huang proposes a neural network and image processing based application to detect and classify the quality of acera nuts. The defects in the areca nuts are segmented using a detection line method. 6 geometric and 3 color features are used to classify the nuts using a neural network. The method achieves an accuracy of 90.9% [33].

Fig. 2. Illustration of inter-class similartity (Stable frame of video CV19_video_4.avi). The red dice is similar to the hazelnut in an under-exposed condition illustrating the inter-class similarity challenge.

A computer vision based machine for walnut sorting has been developed by Tran et al., in 2017. The system uses normalized histogram features and SVM for classification of walnut followed by conveyor systems to sort the walnut [34].

A Linear Discriminant Analysis based classifier using the time series acoustic data of hazelnut is used to classify the infected kernels. The sub-bands of the time series data is obtained using the wavelet transform, and this serves as the features for the classifier [35].

A vegetable and fruit recognition system is developed using Convolutional Neural Networks (CNN) [36]. The paper trains ResNet and NASNet architectures on the VegFru dataset and achieves state-of-the-art classification accuracy. However, localization is unexplored in this system.

A work on detecting and classifying hazelnut is proposed in [37]. The research obtains real-time inference using mean-based classification and K-means clustering. The method achieves an accuracy above 90%. However, the method requires a uniform background without any other object classes and requires proper illumination conditions.

An intelligent peanut classification model is developed by Narendra V G, et al., [38]. The study utilizes color and texture features to classify a variety of peanuts. The study performs a comparative analysis of various machine learning models in the classification task, namely, Random Forest (RF), Multi-layer Perceptron (MLP) and libSVM. libSVM outperforms other machine learning models.

The previous image processing and learning based works on edible nuts clearly provide no signs of both localization and classification of the three types -peanut, walnut, and hazelnut in a single system. In addition, the systems focussing on localization and classification of a single nut type are effective only in a restricted environment without many distractor objects in the scene. Therefore, the proposed work focus on the detection of peanut, walnut, and hazelnut in a video input. The detection is robust to variations in illumination conditions, number of distractors, viewpoints, and frame rate of the video.

IV. Proposed strategy

The proposed strategy is illustrated in figure 3. This is a hybrid approach integrating classical approaches and deep learning. The classical approach is used to extract the stable frame and enhance the image contrast for detection using a deep architecture. The deep learning based object detection requires numerous training data. The available CV 2019 dataset included 380 videos. Each video provides a stable frame resulting in 380 training images. However, it is less given the variations to learn in the video data and the demanded robustness of the application. Out of the available 380 images, 30 randomly chosen images in the video package 1_1 and video package 2_2[1] of the CV 2019 dataset is used for testing. Video package 1_1

[1]Video package 1_1- video package 1 part 1. Similarly, video package 2_2- video package 2 part 2.

and 2_2 is considered because the videos are relatively challenging with large variations compared to other video packages. The 350 images of the remaining video data are used to generate a synthetic dataset of 5000 images. The synthetic dataset is divided into training and validation set for training the Mobilenet-SSD object detector. In the application part of the pipeline, for a new video data given, CLAHE is applied to the stable frame extracted. CLAHE is a histogram equalization method performing contrast enhancement. It is applied because the dataset includes lesser under-exposed images, as shown in 2. The trained model is used for prediction in the application phase. The results are written to a CSV file with an entry for each nut detected. Each entry includes a tuple with the stable frame extracted in the video, a point (x,y) with the coordinate value of each nut in the stable frame, and the corresponding class name (nut type).

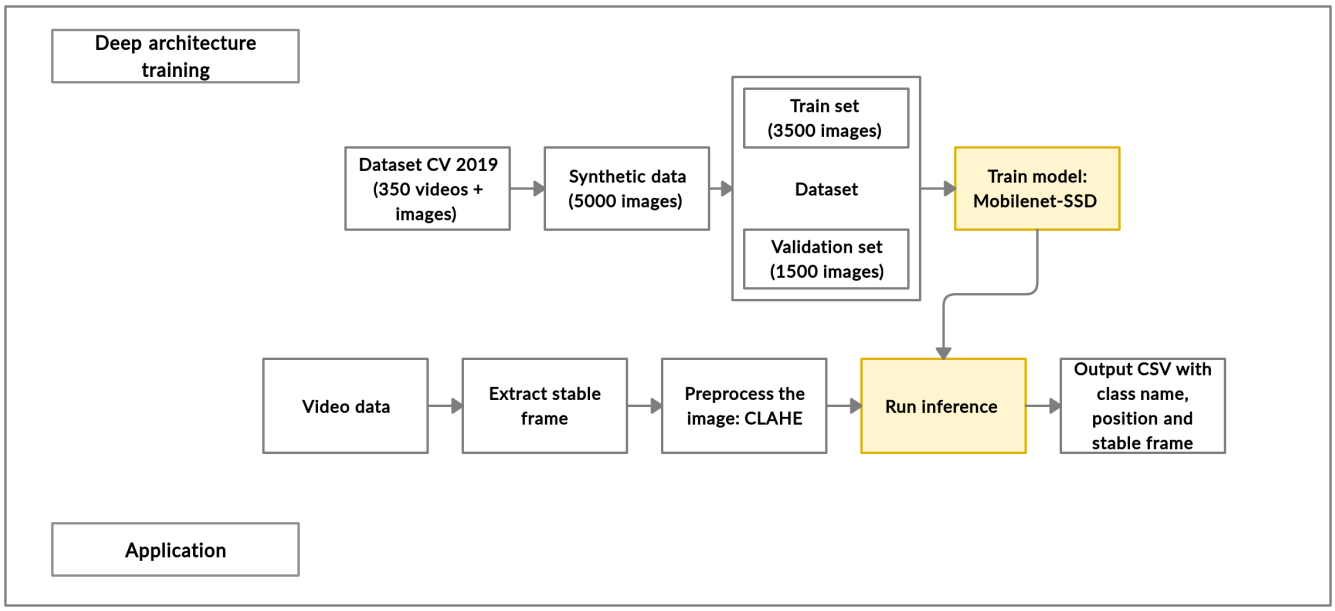


Fig. 3. Proposed strategy to detect the three classes of nuts. The pipeline is divided into two parts- Deep architecture training and application. The yellow coloring indicates the usage of the same model trained for prediction in the application part of the pipeline. The output Comma Separated Values file (CSV) includes an entry with the stable frame, x and y position on the nut and the class label of the nut.

V. Materials

This section provides an overview of the various materials and methods used in the proposed strategy. The following materials are used in the project at various stages of the proposed pipeline given in figure 3- Python 3.5, OpenCV 3.4.2, Tensorflow 1.13.2, LabelMe 4.1.1, Pyinstaller 3.6, Numpy 1.16, Pandas 0.25, Scikit learn 0.22 and HBRS cluster[2].

VI. Approach

This section provides an overview of the approach followed in the project to detect nuts. The methods are supported by the illustration of outputs at each stage of the process.

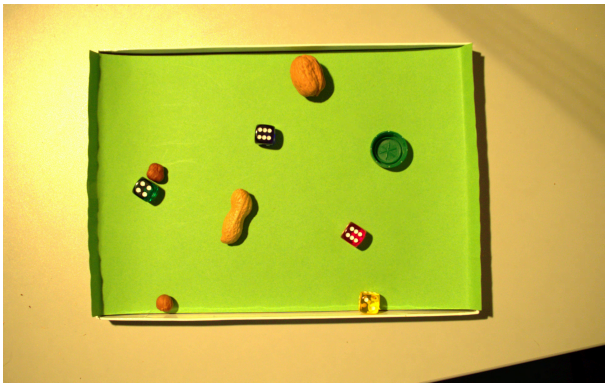
A. Frame acquisition

The stable frame acquisition is carried out using frame differencing. In order to reduce the computation time and memory, this method adopts a mechanism by selecting only two frames per second of the video.

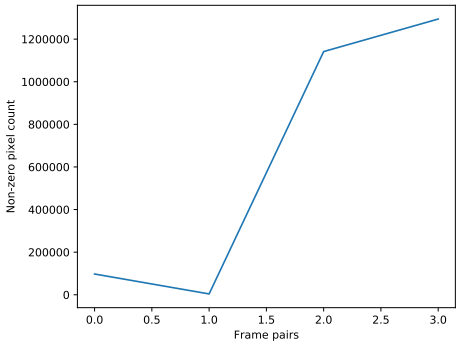
[2]The specifications of the cluster is available at https://wr0.wr.inf.h-brs.de/wr/index.html

One of the assumptions is that the objects are stationary for at least half a second in the video. Thus, all frames are read, and the frame number of each frame is saved in a separate list. The first few frames and the last few frames are ignored to make the frame extraction step focus only on the middle part of every video. This approach is considered because most of the videos start with a hand about to throw the dice on the tray and ends with a hand collecting all the dice back from the tray.

The sampled frames are stored in a list after converting them into grayscale using *cvtColor* method of OpenCV. The images are converted to grayscale because the complexity of the model is less and easy to process. The frames in the list are considered in pairs from the beginning to the end. The absolute difference between every pair of frames is acquired as an image using OpenCV *absdiff* method. Following this, the image is processed using a Gaussian blur of 3 by 3 kernel and binary thresholded to remove the slight movement of any background elements. Finally, the number of non-zero/white pixels in the final image obtained. The list of the number of non-zero pixels for every pair of frames represents the amount of movement that occurred between the two frames in every pair. Thus, the second frame of the pair having minimum movement between the frames represents no movements. This second frame is the stable frame. Figure 4 shows the plot of frame difference between every pair of frames for one of the videos, ”CV19_video_57.avi”. As evident from the plot, the frame corresponding to the 2nd pair represents the frame with minimum movement.



(a) Stable frame extracted.



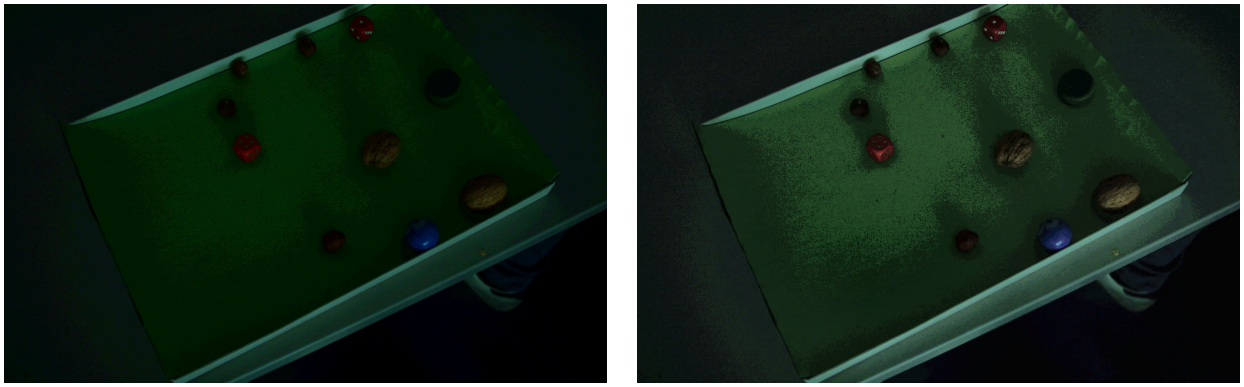
(b) Graph of the non-zero pixels versus the frame pairs sampled.

Fig. 4. Illustration of frame differencing implemented on video CV19_video_57.avi. The frame number corresponding to the second frame of frame pair 1 is selected.

B. Contrast enhancement

The dataset is biased with less number of low contrast and under-exposed images. To overcome this issue, the contrast enhancement of the stable frame is carried out using CLAHE. Ordinary AHE tends to overamplify the contrast in near-constant regions of the image since the histogram in such regions is highly concentrated. As a result, AHE may cause noise to be amplified in near-constant regions. CLAHE is a variant of adaptive histogram equalization in which the contrast amplification is limited, so as to reduce this problem of noise amplification. In CLAHE, the contrast amplification in the vicinity of a given pixel value is given by the slope of the transformation function [30]. The following figure 5 illustrates the CLAHE processed image. The CLAHE is done using the following steps adapted from the work [31].

1) Convert RGB image to LAB color space.
2) Apply CLAHE to L-channel.
3) Merge the CLAHE enhanced L-channel with the a and b channel.
4) Convert back to the RGB model.



(a) Stable frame extracted.

(b) CLAHE output with relatively better contrast than the stable frame.

Fig. 5. Illustration of CLAHE processed stable frame of video CV19_video_4.avi. The CLAHE enhanced image is apparantly better showing the characteristics of walnut and distractor (right side dark green bottle cap in the image).

C. Synthetic dataset

The synthetic dataset is generated using the artificial image generator tool developed by [4]. 350 images from CV 2019 dataset are used to generate 5000 synthetic images. The need for more training data is to include the various conditions of the problem in the training data such as illumination, viewpoints, and a number of distractors. This helps in training the network and generalization performance of the network [39]. The images are generated by randomly placing objects on a variety of different background images automatically. The inputs to the tool are the image with a single nut and boundary of the nut with the class label (semantic label). The tool alters the position, scale, and orientation of the nut provided in the image on random background images. The final number of images to be generated by the tool can be passed to the tool as a command-line argument. The object detection Pascal VOC format labels are also provided by the tool.

Fig. 6. Example images from the synthetic dataset generated. The nuts are extracted from the stable frame ground truth images provided in the CV 2019 dataset (The background images are taken from http://wallpaperswide.com).

D. Training and inference using MobileNet-SSD

The synthetic dataset is divided into 3500 images for training and 1500 images for validation. MobileNet-SSD is the model selected for training. MobileNet-SSD offers superior performance on the GPU used for inference. The test GPU is NVIDIA 1050Ti because of the model compression network, MobileNet [28]. This provides a robust detection system.

The training is carried out using the TensorFlow object detection API. The train and validation record files are created for the train and validation set. The training is performed until the validation error, and train error coincide. The training batch size is 64, and training is performed for 10000 steps.

The stable frame after preprocessing using CLAHE is resized to 300 x 300 pixels. This is the input size of MobileNet-SSD. MobileNet-SSD is also available at 512 x 512 pixels input. However, 300 x 300 pixels is chosen to reduce computation cost. The output of the inference is the class names and the bounding box of the respective class. The bounding box is rescaled to match the actual image size. An example inference result is shown in figure 7.

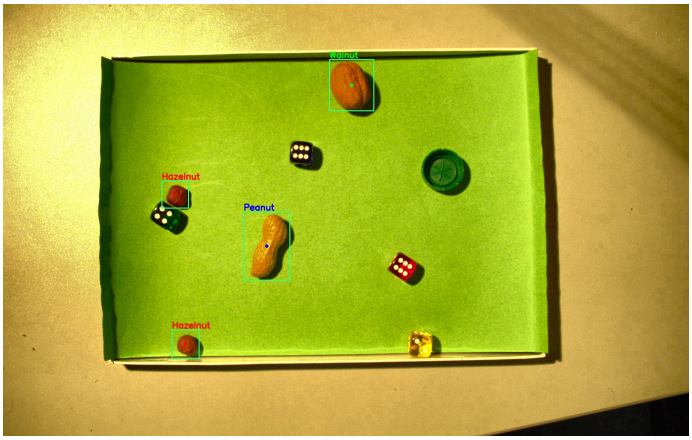
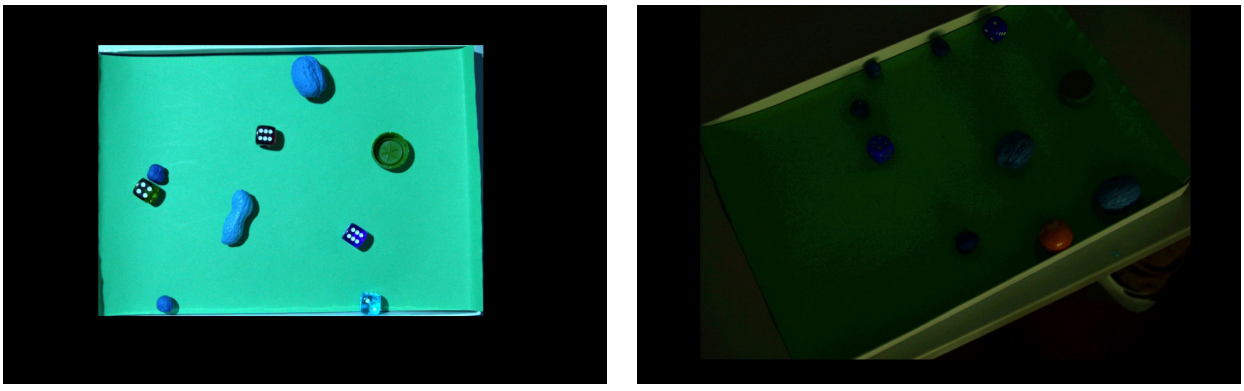


Fig. 7. Inference result for CV19_video_57.avi. All the objects are detected precisely. The box around the nuts are the bounding box with class label near the box.

E. Extracting Region of Interest (ROI)

The region of interest in this task is the rectangular box containing the nuts and distractors in the stable frame. To extract the region of interest, initially, K-means clustering is used as the nuts are rolled on a uniformly colored rectangular box. K-means clustering is performed to segment the background from the rectangular box of interest. The figure 8 illustrate the output images obtained by K-means clustering.



(a) K-means result on the stable frame of CV19_video_57.avi.

(b) K-means result on the stable frame of CV19_video_4.avi.

Fig. 8. Results of K-means clustering performed (illustration purpose only, not included in the pipeline). As K-means clustering increased the inference time of the proposed application, K-means is not implemented in the pipeline.

However, due to increased inference time on including K-means in the pipeline, a workaround method is implemented in the pipeline by labeling the box and training the model to detect the rectangular box. The predictions inside the rectangular box are filtered to provide the results as shown in figure 9

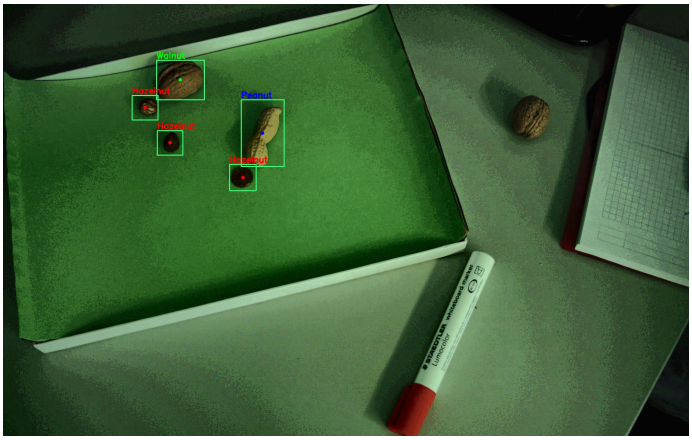


Fig. 9. Illustration of performing detection inside the rectangular box using the video CV19_video_199.avi. In this figure, a walnut outside the box is not detected.

VII. Results and discussion

This section provides the training results of MobileNet-SSD with the synthetic dataset created, stable frame extraction algorithm accuracy, and the evaluation of the DeepNut detector developed. In addition, the limitations of the DeepNut detector are discussed. The evaluation is performed on 15 test videos. Out of 15 test videos, 14 videos are randomly selected from video package 1_1 and one video from video package 2_2 where the stable frame extraction failed.

A. MobileNet-SSD performance on validation set

The performance of the MobileNet-SSD model is evaluated on the 1500 images of the validation set at 0.5 Intersection over Union (IoU). IoU is the ratio of the area of overlap between the ground truth bounding box, and the model predicted bounding box. The reported mean Average Precision (mAP) is provided in table I.

TABLE I
PERFORMANCE OF SSD- MOBILENET MY-TABLEON VALIDATION SET.

mean Average Precision (mAP)	Intersection over Union (IoU)
0.8812	IoU >0.5

Table I: Intersection over union is the ratio of area of overlap over the area of union in the detection provided. The mAP is a measure of the classification of DeepNut detector at the provided IoU. The metrics is provided on the validation set with 1500 images including the synthetic images and images provided in the CV 2019 dataset.

B. Stable frame extraction

The stable frame extraction method is tested on the 30 test videos randomly chosen from the video package 1_1 and video package 2_2. As mentioned before, the video package 1 is chosen because it includes a wide variety of videos relative to other packages by observation. The stable frame is the frame where all the objects are static. The image of the stable frame is checked against the ground truth image provided by measuring the Euclidean distance of the histogram features of both images with 256 bins.

$$\text{Stable frame extraction accuracy} = \frac{\text{Number of correctly extracted stable frames}}{\text{Number of videos tested}} \times 100 \qquad (1)$$

$$\text{Stable frame extraction accuracy } (\%) = \frac{14}{15} \times 100$$
$$\text{Stable frame extraction accuracy} = 93.33\%$$

TABLE II
STABLE FRAME EXTRACTION EVALUATION

Serial no.	Video number	Stable frame correctly extracted	Score
1	CV19_video_4.avi	Yes	1
2	CV19_video_5.avi	Yes	1
3	CV19_video_12.avi	Yes	1
4	CV19_video_54.avi	Yes	1
5	CV19_video_57.avi	Yes	1
6	CV19_video_73.avi	Yes	1
7	CV19_video_88.avi	Yes	1
8	CV19_video_89.avi	Yes	1
9	CV19_video_115.avi	Yes	1
10	CV19_video_136.avi	Yes	1
11	CV19_video_198.avi	Yes	1
12	CV19_video_199.avi	Yes	1
13	CV19_video_366.avi	No	0
14	CV19_video_55.avi	Yes	1
15	CV19_video_66.avi	Yes	1
	Total score		14

Table II: The stable frame extraction is considered to be correctly extracted if the stable frame extracted and the ground truth provided for each video matches. Each correct stable frame extraction is awarded 1 point. 14 videos are randomly selected from video package 1_1 and one video from video package 2_2 where the stable frame extraction failed.

The single failed case includes a black background with an under-exposed lighting condition, as shown in figure 10.

(a) Ground truth stable frame provided for CV19_video_366.avi.

(b) Extracted stable frame of CV19_video_366.avi by the frame acquisition algorithm employed.

Fig. 10. Failure of stable frame extraction of video CV19_video_366.avi. The stable frame extraction method fails on dark background with a dark lighting condition and the video is at a slower frame rate.

C. Evaluation of the DeepNut detector

This paper performs the detection of nuts integrating nut localization followed by nut classification.

1) Localization task: The evaluation of the localization task includes suggesting a point on the nut with a correct label. The correct and wrong localizations are awarded '1 point'. However, the correct and wrong localizations are subtracted to find the effective localization score. Finally, the total score obtained is calculated over the total achievable score in terms of percentage. The table III summarize the scores over 15 test videos.

The efficiency of localization is given by the following equation.

$$\text{Localization efficiency } (\%) = \frac{\text{Number of nuts located - Number of wrong and/or missed localizations}}{\text{Number of total nuts in all videos}} \times 100 \tag{2}$$

TABLE III
EVALUATION OF DEEPNUT LOCALIZATION

Serial number	Video number	Number of nuts in the video	Number of nuts located	Number of wrong and/or missed localizations
1	CV19_video_4.avi	6	5	3
2	CV19_video_5.avi	8	8	0
3	CV19_video_12.avi	9	9	0
4	CV19_video_54.avi	8	8	0
5	CV19_video_57.avi	4	4	0
6	CV19_video_73.avi	5	5	0
7	CV19_video_88.avi	10	10	0
8	CV19_video_89.avi	10	10	0
9	CV19_video_115.avi	1	1	0
10	CV19_video_136.avi	9	7	2
11	CV19_video_198.avi	4	4	0
12	CV19_video_199.avi	4	4	1
13	CV19_video_366.avi	6	0	0
14	CV19_video_55.avi	3	3	0
15	CV19_video_66.avi	3	3	0
	Total score	90	81	6

Table III: The number of nuts is total nuts in the stable frame in the ground truth stable frame image provided for each video. The number of nuts located is the number of correct localization by the DeepNut detector. The number of wrong and/or missed localizations include the number of nuts not detected or spurious detection such as locating a distractor as nut or showing random localizations in the stable frame extracted. The videos are randomly selected as stated in the beginning of section VII.

$$\text{Localization efficiency } (\%) = \frac{75}{90} \times 100$$
$$\text{Localization efficiency } = 83.33\%$$

2) Classification task: The following terminologies are used in the evaluation metrics of the DeepNut detector developed.

Each classification by a classifier model is divided into 4 categories namely, True positive, True negative, False positive, and False negative. Each category is given *1 point*. Finally, the metrics are computed using these values.

True positive (TP) for a particular data point represents the positive prediction of a positive sample by the classifier model. True negative (TN) represents the negative prediction of a negative data point by the model. False positive (FP) represents the positive prediction by a model for a negative data point. False negative (FN) represents the negative prediction by a model for a positive data point. Therefore, TP and TN are correct predictions by the classifier. In contrast, FP and FN are wrong predictions by the classifier.

The DeepNut is evaluated using the following classification metrics:

Precision provides the ratio of positive samples correctly classified to the total positive samples in the test set. It is a measure of the exactness of the classifier. "What proportion of positive predictions by the classifier model is actually positive?" [32]. The model predicting zero FP has a precision value 1.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{3}$$

Recall provides a measure of positive samples correctly classified to the total positive predictions by the classifier model. It is a measure of the completeness of the classifier and called sensitivity. "What proportion of actual positive sample is predicted correctly by the classifier model?" [32]. The classifier model predicting zero FN has a recall value 1.

$$\text{Recall } = \frac{TP}{TP + FN} \tag{4}$$

F1 score provides a balance between precision and recall taking note of the false negative and false positives. It is a better metric than accuracy in imbalanced datasets.

$$\mathrm{F1} = 2 \times \frac{\text{Precision } \times \text{Recall}}{\text{precision } + \text{Recall}} \tag{5}$$

The tables IV, V, and VI provides the TP, FP, and FN for peanut, walnut, and hazelnut respectively.

TABLE IV
EVALUATION OF DEEPNUT CLASSIFICATION FOR PEANUT

Serial number	Video number	TP	FP	FN
1	CV19_video_4.avi	0	0	0
2	CV19_video_5.avi	2	0	0
3	CV19_video_12.avi	2	0	0
4	CV19_video_54.avi	3	0	0
5	CV19_video_57.avi	1	0	0
6	CV19_video_73.avi	1	0	0
7	CV19_video_88.avi	3	0	0
8	CV19_video_89.avi	3	0	0
9	CV19_video_115.avi	1	0	0
10	CV19_video_136.avi	2	0	1
11	CV19_video_198.avi	1	0	0
12	CV19_video_199.avi	1	0	0
13	CV19_video_366.avi	0	0	2
14	CV19_video_55.avi	1	0	0
15	CV19_video_66.avi	0	0	0
	Total score	21	0	3

Table IV: TP, FP and FN are True Positive, False Positive and False Negative, respectively (definition provided in section Classification task VII-C2). TP- peanut present in the stable frame is correctly classified, FP- peanut is unavailable in the stable frame but DeepNut classifies a walnut, hazelnut or distractor as peanut, FN- peanut is available in the stable frame and DeepNut classifies the peanut wrongly. Every metric is given 1 point. The videos are randomly selected as stated in the beginning of section VII.

TABLE V
EVALUATION OF DEEPNUT CLASSIFICATION FOR WALNUT

Serial number	Video number	TP	FP	FN
1	CV19_video_4.avi	1	0	1
2	CV19_video_5.avi	3	0	0
3	CV19_video_12.avi	4	0	0
4	CV19_video_54.avi	3	0	0
5	CV19_video_57.avi	1	0	0
6	CV19_video_73.avi	0	1	0
7	CV19_video_88.avi	4	0	0
8	CV19_video_89.avi	2	0	0
9	CV19_video_115.avi	0	0	0
10	CV19_video_136.avi	3	0	0
11	CV19_video_198.avi	2	0	0
12	CV19_video_199.avi	1	0	0
13	CV19_video_366.avi	0	0	1
14	CV19_video_55.avi	1	0	0
15	CV19_video_66.avi	0	0	0
	Total score	25	1	2

Table V: TP, FP and FN are True Positive, False Positive and False Negative, respectively (definition provided in section Classification task VII-C2). TP- walnut present in the stable frame is correctly classified, FP- walnut is unavailable in the stable frame but DeepNut classifies a peanut, hazelnut or distractor as walnut, FN- walnut is available in the stable frame and DeepNut classifies the walnut wrongly. Every metric is given 1 point. The videos are randomly selected as stated in the beginning of section VII.

The table VII summarize the evaluation metrics for peanut, walnut, and hazelnut individually. In addition, the mean F-score of the DeepNut detector is 0.921 in the validation set developed.

The proposed DeepNut detector performance is relatively less on detecting smaller hazelnuts.

TABLE VI
EVALUATION OF DEEPNUT CLASSIFICATION FOR HAZELNUT

Serial number	Video number	TP	FP	FN
1	CV19_video_4.avi	4	2	0
2	CV19_video_5.avi	3	0	0
3	CV19_video_12.avi	3	0	0
4	CV19_video_54.avi	2	0	0
5	CV19_video_57.avi	2	0	0
6	CV19_video_73.avi	1	2	0
7	CV19_video_88.avi	3	0	0
8	CV19_video_89.avi	5	0	0
9	CV19_video_115.avi	0	0	0
10	CV19_video_136.avi	2	0	1
11	CV19_video_198.avi	1	0	0
12	CV19_video_199.avi	2	0	1
13	CV19_video_366.avi	0	0	3
14	CV19_video_55.avi	1	0	0
15	CV19_video_66.avi	3	0	0
	Total score	32	4	4

Table VI: TP, FP and FN are True Positive, False Positive and False Negative, respectively (definition provided in section Classification task VII-C2). TP- hazelnut present in the stable frame is correctly classified, FP- hazelnut is unavailable in the stable frame but DeepNut classifies a peanut, walnut or distractor as hazelnut, FN- hazelnut is available in the stable frame and DeepNut classifies the hazelnut wrongly. Every metric is given 1 point. The videos are randomly selected as stated in the beginning of section VII.

TABLE VII
EVALUATION METRICS OF DEEPNUT CLASSIFICATION

Metric	Peanut	Walnut	Hazelnut
Precision	1	0.961	0.888
Recall	0.875	0.926	0.888
F-score	0.933	0.943	0.888

Table VII: Precision, Recall and F-score is provided for each class. The calculations use the formula provided in section VII-C2 and values from tables IV, V and VI. The validation set with 1500 images is used to evaluate the DeepNut detector. Hazelnut classification is relatively challenging compared to other nuts using the developed DeepNut detector.

D. Discussion: Limitations of DeepNut

In addition to the limitation of the stable frame extraction provided in figure 10. The system fails at the follows two conditions, 1. under-exposed and shadowed environment (refer figure 11) 2. images resembling distortion (refer figure 12).

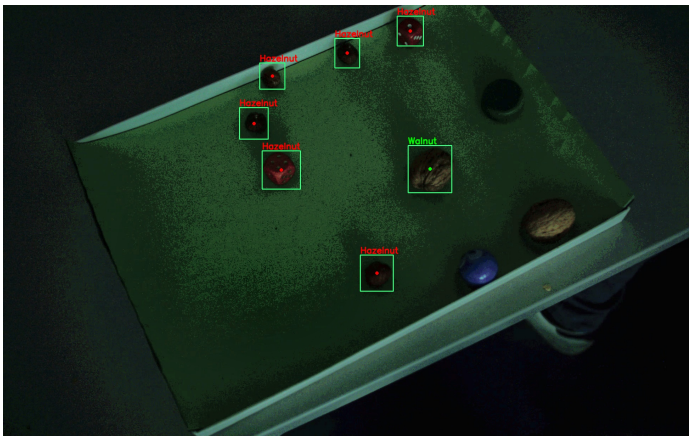


Fig. 11. Inference result for CV19_video_4.avi. The red dice is classified as a hazelnut and the walnut is not identified due to the under-exposed and shadowed environment.

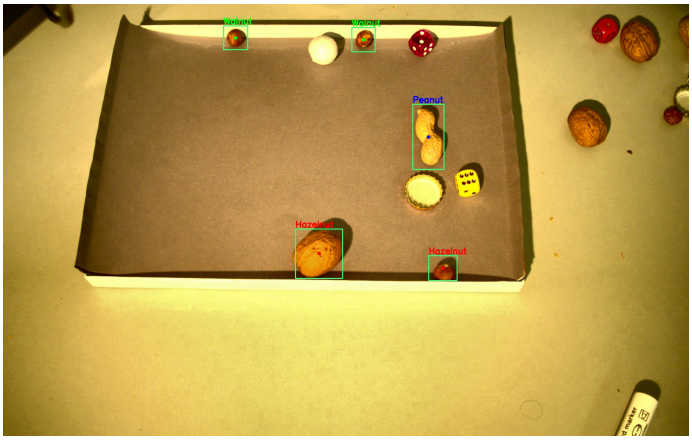


Fig. 12. Inference result for CV19_video_73.avi. The hazelnut at top of this image is misclassified. DeepNut fails at images resembling a distorted image.

VIII. Future work

The primary future work concentrates on classical methods. On drawing inspirations from fellow student competitors, it is understood that classical methods can solve the task. The applicability of classical methods to solve the task has to be studied in detail. However, at the beginning of the project work, the author cannot achieve reasonable generalization error using traditional methods such as template and shape matching. The second prospective future study includes exploring various state-of-the-art deep architectures available. The performance of other deep architectures for this nut detection task can be studied. Finally, in this study, at places of choosing between high accuracy and less inference time, the importance is given to inference time. For instance, SSD with an input size 300 is used in this study because SSD 512 has a relatively higher processing time and higher accuracy. However, the effect of this compromise can be studied in detail.

Acknowledgment

I thank Prof. Dr. Rainer Herpers, who gave the opportunity to work on this challenging task and get our hand dirty by implementing the Computer Vision techniques. I express my gratitude to Christoph Pomrehn, who patiently supported us during the entire course of this project work, from recording videos to submitting the final executable files. I thank all my fellow students who were involved in recording videos. Finally, I would like to extend my gratitude to my family and friends, whose love and moral support helped me in steering this project at the right pace.

References

[1] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. ”Mobilenets: Efficient convolutional neural networks for mobile vision applications.” arXiv preprint arXiv:1704.04861, 2017.

[2] Zou, Zhengxia, Zhenwei Shi, Yuhong Guo, and Jieping Ye. ”Object Detection in 20 Years: A Survey.” arXiv preprint arXiv:1905.05055, 2019.

[3] Guo, Jiajia, Junping Wang, Ruixue Bai, Yao Zhang, and Yong Li. ”A new moving object detection method based on frame-difference and background subtraction.” In IOP Conference Series: Materials Science and Engineering, vol. 242, no. 1, p. 012115. IOP Publishing, 2017.

[4] Naresh Kumar Gurulingan. Chapter: Dataset creation - artificial image generation algorithm. Github, December 2018. Accessed on: 2019-10-29. [Online]. URL: https://github.com/NareshGuru77/SemanticSegmentation/blob/master/Report/GurulinganNK-RnD-Report.pdf

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection., 2005.

[6] G. Garrido and P. Joshi. OpenCV 3.X with Python By Example - Second Edition: Make the Most of OpenCV and Python to Build Applications for Object Recognition and Augmented Reality. Packt Publishing, 2nd edition, 2018. Chapter: Object tracking- Frame differecing, Accessed on: 2019-10-29. [Online].

[7] Lowe, David G. ”Distinctive image features from scale-invariant keypoints.” International journal of computer vision 60, no. 2, 91-110, 2004.

[8] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In Advances in neural information processing systems, pp. 91-99, 2015.

[9] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[10] de Souza, Rávila, Raquel Schincaglia, Gustavo Pimentel, and João Mota. "Nuts and human health outcomes: A systematic review." Nutrients 9, no. 12 (2017): 1311.

[11] K. Wada. labelme: Image Polygonal Annotation with Python. GitHub https://github.com/wkentaro/labelme, 2016. Accessed on: 2019-10-29. [Online].

[12] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014.

[13] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9, 2015.

[14] Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems, 2019.

[15] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708, 2017.

[16] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[17] Dai, Jifeng, Yi Li, Kaiming He, and Jian Sun. "R-fcn: Object detection via region-based fully convolutional networks." In Advances in neural information processing systems, pp. 379-387, 2016.

[18] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.

[19] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.

[20] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2017.

[21] M. watch. Dried Fruits and Edible Nuts Market 2019. Accessed on: 2019-12-10. [Online], 2019. URL: https://www.marketwatch.com/pressrelease/dried-fruits-and-edible-nuts-market-2019-business-revenue-future-growth-trends-plans-top-key-players-businessopportunities-industry-share-global-size-analysis-by-forecast-to-2024-360-research-reports-2019-08-30.

[22] U. Food and D. Administration. MPM: V-10. Nuts and Nut Products Methods. Accessed on: 2019-12-10. [Online]. 2017. URL: https://www.fda.gov/food/laboratory-methods-food/mpm-v-10-nuts-and-nut-products-methods.

[23] Tran, Truong, Tu Nguyen, Mai Nguyen, and Tuan Pham. "A Computer Vision Based Machine for Walnuts Sorting Using Robot Operating System." In International Conference on Advances in Information and Communication Technology, pp. 9-18. Springer, Cham, 2016.

[24] Cavallaro, Elizabeth, Kashmira Date, Carlota Medus, Stephanie Meyer, Benjamin Miller, Clara Kim, Scott Nowicki et al. "Salmonella typhimurium infections associated with peanut products." New England Journal of Medicine 365, no. 7 (2011): 601-610.

[25] O'Mahony, Niall, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. "Deep Learning vs. Traditional Computer Vision." In Science and Information Conference, pp. 128-144. Springer, Cham, 2019.

[26] Lee, Andy. "Comparing deep neural networks and traditional vision algorithms in mobile robotics." Swarthmore University, 2015.

[27] Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size." arXiv preprint arXiv:1602.07360, 2016.

[28] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861, 2017.

[29] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018.

[30] Karel Zuiderveld. Contrast limited adaptive histogram equalization. Graphics gems IV. Academic Press Professional, Inc., USA, 474–485, 1994

[31] Bora, Dibya Jyoti, Anil Kumar Gupta, and Fayaz Ahmad Khan. "Comparing the performance of L* A* B* and HSV color spaces with respect to color image segmentation." arXiv preprint arXiv:1506.01472, 2015.

[32] Google AI, Machine learning course, 2018. Accessed on: 2019-12-29. [Online]. URL: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

[33] Huang, Kuo-Yi. "Detection and classification of areca nuts with machine vision." Computers Mathematics with Applications 64, no. 5, pp. 739-746, 2012.

[34] Tran, Truong, Tu Nguyen, Mai Nguyen, and Tuan Pham. "A Computer Vision Based Machine for Walnuts Sorting Using Robot Operating System." In International Conference on Advances in Information and Communication Technology, pp. 9-18. Springer, Cham, 2016.

[35] Kalkan, Habil, Nuri Firat Ince, Ahmed H. Tewfik, Yasemin Yardimci, and Tom Pearson. "Classification of hazelnut kernels by using impact acoustic time-frequency patterns." EURASIP Journal on Advances in Signal Processing 2008, no. 1, 1-11, 2007.

[36] Buzzelli, Marco, Federico Belotti, and Raimondo Schettini. "Recognition of Edible Vegetables and Fruits for Smart Home Appliances." In 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), pp. 1-4. IEEE, 2018.

[37] Solak, Serdar, and Umut ALTINIŞIK. "Detection and classification of hazelnut fruit by using image processing techniques and clustering methods." Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi 22, no. 1, 56-65, 2018.

[38] V G, Narendra, Anita S. Kini, Anita S. Kini. "An intelligent classification model for peanut's varieties by color and texture features." International Journal of Engineering Technology [Online], 7.2.27 (2018): 250-254, 2020
[39] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.