

# The nut job

1<sup>st</sup> Priteshkumar Gohil *Department of Computer Science  
Hochschule Bonn-Rhein-Sieg  
Bonn, Germany  
priteshbgohil@gmail.com*

## Abstract

Object detection is the task in computer vision to make computer intelligent to recognize and locate the objects. Some of the domains in the object detection are less explored and one such is the food domain. Food packaging industry e.g. peanut packaging service might require to detect bad quality nuts. This paper presents an approach to detect nuts in the scene with cluttered objects, varying illumination condition, and static images. We use Convolution Neural Network (CNN), a deep learning approach to learn features, classify and localize nuts in the scene. Nuts to detect are peanut, walnut and hazelnut. This paper also presents an algorithm to detect stable frames in the video input, which are based on colour histograms and their difference. The frame detection algorithm can correctly detect the stable frame with an accuracy of 98.94%. Our deep learning approach can detect nuts in different challenging situations.

## Index Terms

Object Detection, Deep Learning, Event Detection, Convolution Nerual Network

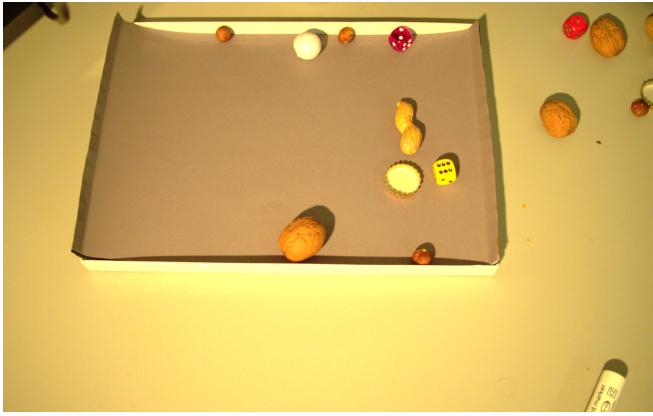
## I. INTRODUCTION

The visual information that humans and animals perceive has played an important role in interacting with their environment. Giving such capabilities to machines and computer to extract the information of interest from the image helped many sectors like healthcare, driving, robotics, food and beverage etc. Therefore, extracting information of interest from images is the central focus of this project.

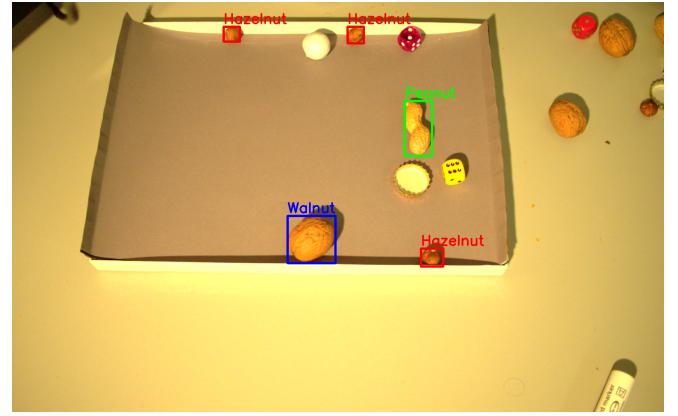
Computer vision addressed the problem of automatically extracting useful information from images such as segmentation, classification and object detection, pattern recognition, and many more. One of the active fields of research is object detection aiming to find the answer of two questions. 1. Which objects are in the image, also known as classification and 2. Where objects are located in the image, also known as localization.

Objects in the image are classified based on the features extracted from the bounding box. Figure 1a and 1b shows the original image and ground truth bounding box. The motivation for selecting features and not pixels for the classification are many. First, features extracted from the object of interest can encode the domain knowledge with finite data. Second, object detection process is faster while using feature based system [1]. Some of the widely used feature extraction techniques include, Histogram of Oriented Gradients (HOG) [2], Speeded-Up Robust Features (SURF) [3], haar wavelet [1] and convolution using CNN [4].

Detecting objects of interest in the real-world such as peanut, walnut and hazelnut pose different challenging situations. Fundamental characteristic of the nuts is its shape, colour, texture and size. For example, nuts in our experiment shares similar hue but slightly different saturation and value. The texture features of the peanut and walnut can easily be confused by the classifier. Also, shape of peanut might look same but the width and height might vary. Desipte challenges from the object properties, camera and surrounding illumination plays an important role. The camera and the colour information from it impose several challenges. Figure 2 illustrates the different challenging situations such as low illumination, high illumination, shadow, occlusion and nuts outside the experimental tray in figure 1a.



(a) Image to detect nuts inside the rectangle tray.



(b) Image annotated with bounding box.

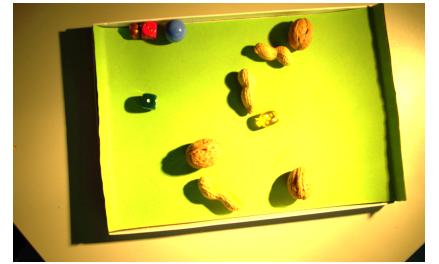
Fig. 1: Input image (left) and the ground truth annotation (right) to show the object of interest.



(a) Poor illumination



(b) High illumination



(c) Object shadow



(d) Cluttered objects

Fig. 2: Images from the dataset.

## II. RELATED WORKS

Kuo-Yi Huang [5] detects the areca nuts using computer vision and neural network techniques. Different features such as geometric properties, defect area and colour information are extracted manually. Then Multilayer perceptron is trained on stack of the extracted features to detect the quality. The author is able to detect the areca nuts with 90.9% accuracy. Similarly, Anita kini et al. [6] detects different texture features from gray level co-occurrence matrix (GLCM) to classify type of peanut.

In [7], deep learning based classifier is designed to detect the quality of pistachios. They use encoder-decoder (autoencoder) structure, where input is first encoded to hidden compressed representation and decoder will try to reproduce an image from the compact hidden representation. They are able to detect bad quality pistachios from oily and darsk stain with an accuracy of 80.3%. However, their approach is not explaining enough about training data and limited to detect only pistachio.

### III. MATERIAL AND METHODS

#### A. Video acquisition

Videos for the dataset was captured in computer vision lab using machine vision system. The system includes camera connected to the computer. A program is written using OpenCV library to capture the video and stable video frame. Each video frames are of  $1936 \times 1216$  pixels and stored in .avi format. Each nuts in the stable frame is annotated with polygon and label using *labelme*<sup>1</sup> tool. Annotations are saved in JSON format. Figure 3 shows the annotation mask on the image with the label of each nuts of interest. Note that only single captured video frame is annotated. We have a total of 380 experiments containing videos, images, and annotations file.

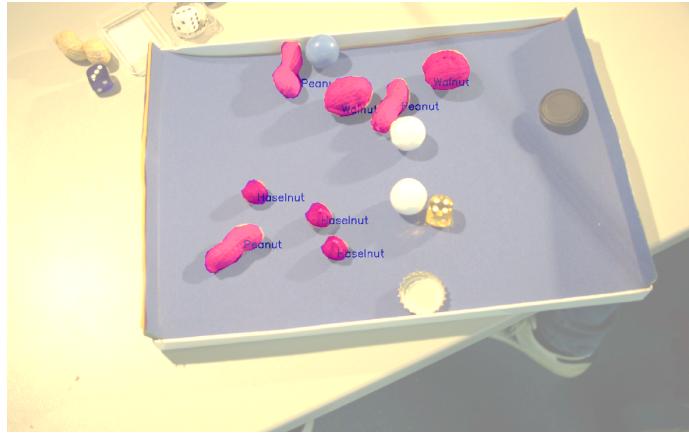


Fig. 3: Image with polygon annotation mask and label.

#### B. Development tools

Following tools and systems are used to detect the nuts in our approach.

- **Operating system:** Ubuntu 16.04
- **Programming language:** Python v3.6
- **Deep learning framework:** Pytorch
- **Training model:** University cluster<sup>2</sup>
- **Testing model:** On Windows 10 and Ubuntu 16.04
- **Executable generator:** Pyinstaller

#### C. Proposed approach

Figure 4 shows the training and testing phase of nut detection. Each module in the training and testing phase is discussed.

##### 1) Training:

**Images and annotation :** Annotated images are being used to train the network because ground truth is very important for training. To prepare training pipeline, each images are resized to  $416 \times 264$  and stored in separate folder. One of the reason to specific resize is the input requirement for You Look Only Once v3 (YOLOv3) and maintaining aspect ration. Whereas, bounding box for each object is extracted from the annotation file, it was normalized and stored in the text file. Each image have label file containing name of the nut, *x*, *y*, *width* and *height*. We split a total of 380 experiments into 80:20 split and have 304 training data and 76 test data.

<sup>1</sup><https://github.com/wkentaro/labelme>

<sup>2</sup><https://wr0.wr.inf.h-brs.de/>

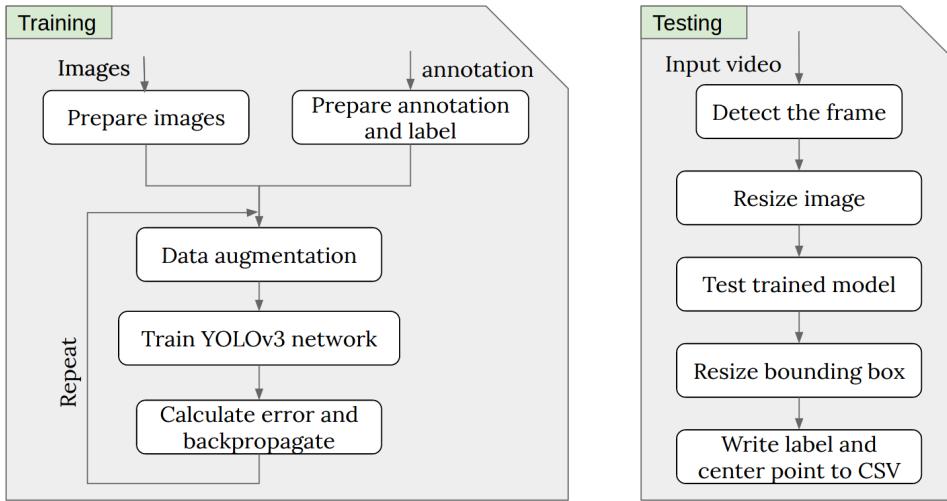


Fig. 4: Training and testing phase of the nuts detection.

**Data augmentation :** Randomly selected input images and bounding boxes are flipped horizontally. Since our dataset contains all the types of variation, we believe that additional augmentations are not required. However, adding other augmentation techniques is the work we would like to do in the future.

**Feature extraction and classification :** We use YOLOv3[8], a state of the art architecture to detect the nuts. The architecture of YOLOv3 can be seen in the figure 5. The maximum input size accepted

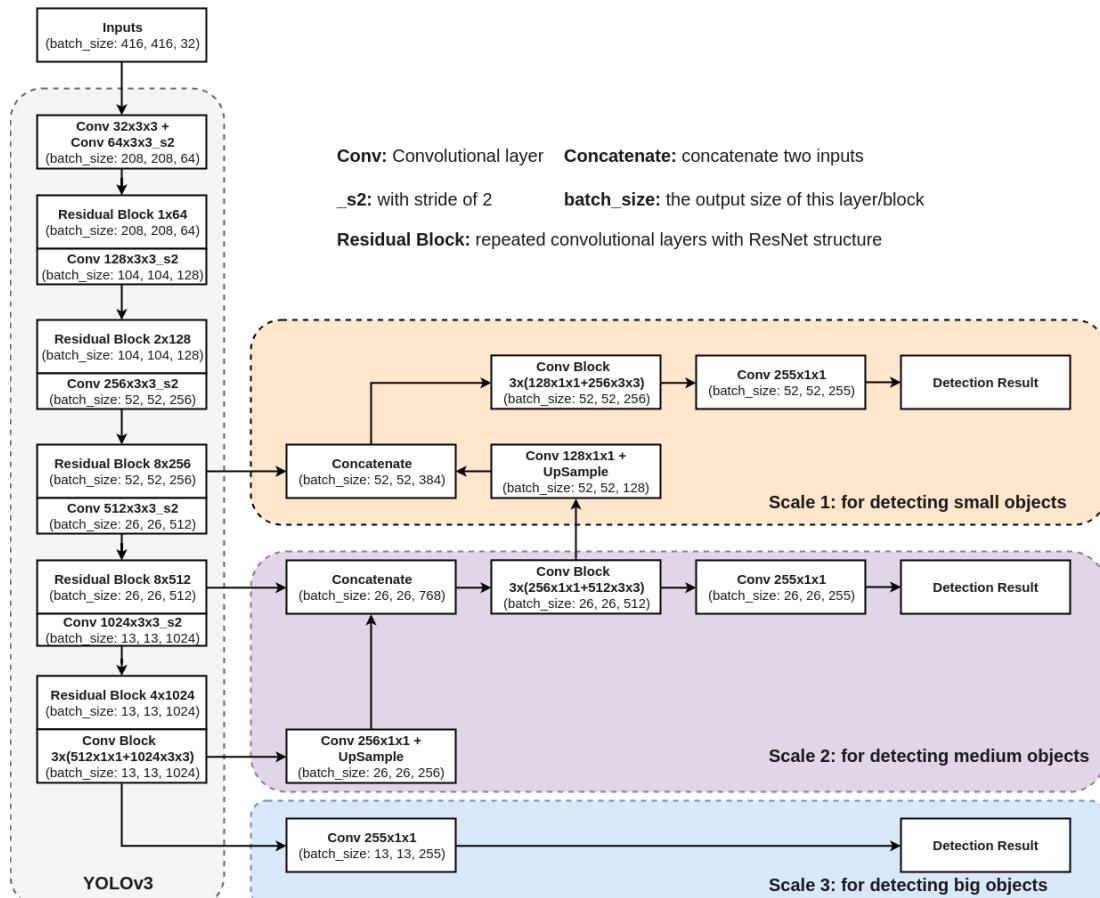


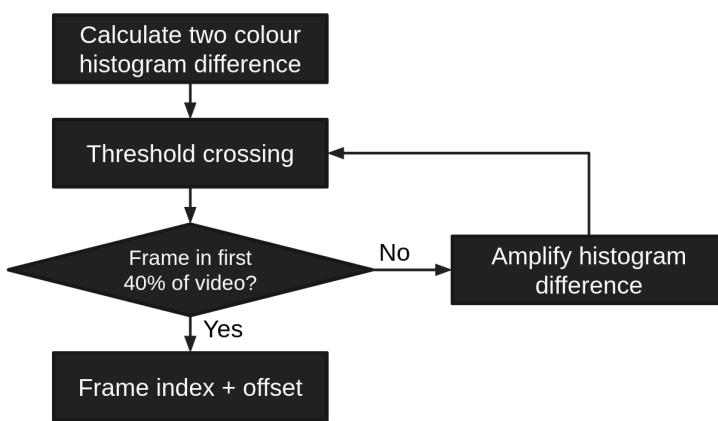
Fig. 5: Architecture of YOLOv3 (image reproduced from [9]).

is  $416 \times 416 \times 3$ . It contains a total of 53 convolution layers. Features from the last convolution layers are flattened and passed through the fully connected layers. Final layer contains 7 neurons in which four values are for bounding box to localize the object, one for confidence of detection and three for each nuts to classify. We have selected batch size of 10 and trained for 900 epoch.

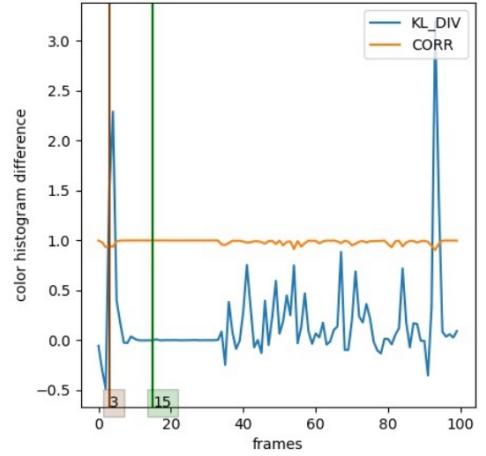
**Error calculation :** Binary cross entropy loss function is used for multiclass binary classification. And mean square error is used to calculate localization error.

2) **Testing:** Our testing phase directly accepts the video input and detects the stable frame with nuts. Detected frame then resized to detect objects using trained model. The output bounding box is then resized to actual input image size and the center point of each bounding box with label is written in specified format in CSV file.

**Stable frame detection :** The stable frame in the video is detected by calculating colour histogram. Figure 6a shows the flow chart of frame detection algorithm. This algorithm works with an assumption that stable frame is in the first 40% of the video frames. The colour histogram of current and previous frame is calculated using OpenCV function cv2.calcHist(). Then difference between two histogram is calculated with two different metrics, correlation difference (cv2.HISTCMP\_CORREL) and KL divergence difference (cv2.HISTCMP\_KL\_DIV). Figure 6b shows the difference of two colour histogram over the entire video. So we try to detect the threshold where KL\_DIV difference is crossing the CORREL difference in first 40% of frames. The flat region in the KL\_DIV difference shows the stable frames. If the threshold is crossing after 40% of frames as can be seen in the figure 7b, KL\_DIV difference is amplified by the factor of 10. This process is repeated until the threshold crosses or 20 iteration, whichever is first. Therefore, we obtain the brown vertical line indicating first highest change in the video. We add offset of 12 to get stable frames (green vertical line) which works for most of the video in our dataset.



(a) Flowchart of stable frame detection algorithm.



(b) Plot of histogram difference.

Fig. 6: Frame detection algorithm (left) and colour histogram difference for video#3 (right).

#### IV. RESULTS

Some of the results of a frame detection algorithm is shown in the figure 7. Whereas, results of the object detection on test data is shown in figure 8.

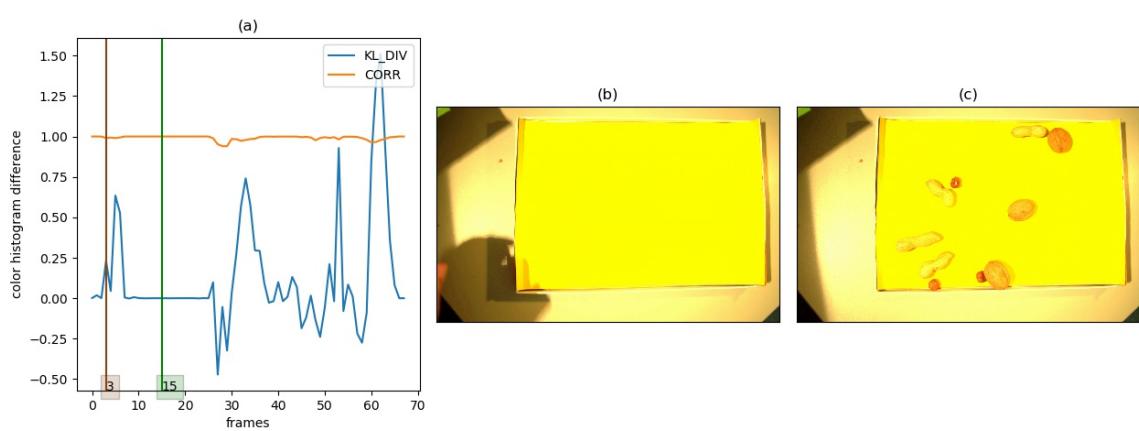
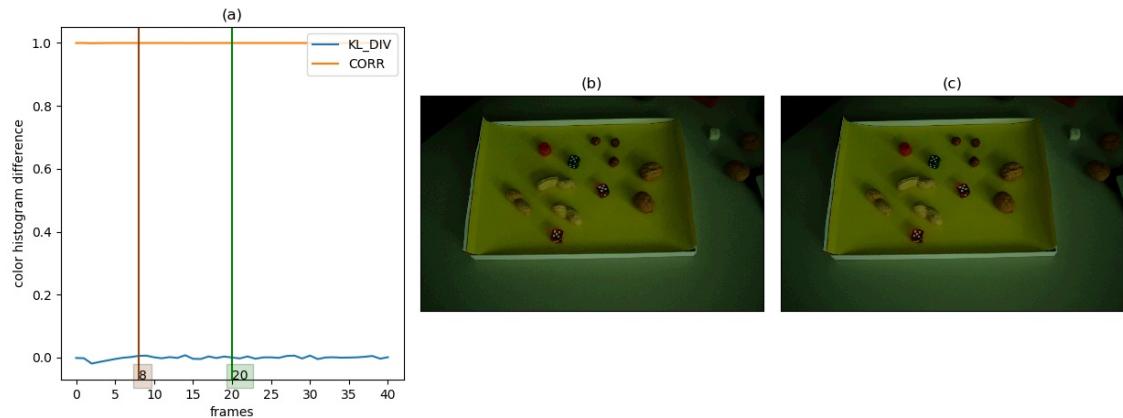


Fig. 7: Frame detection examples.

V.

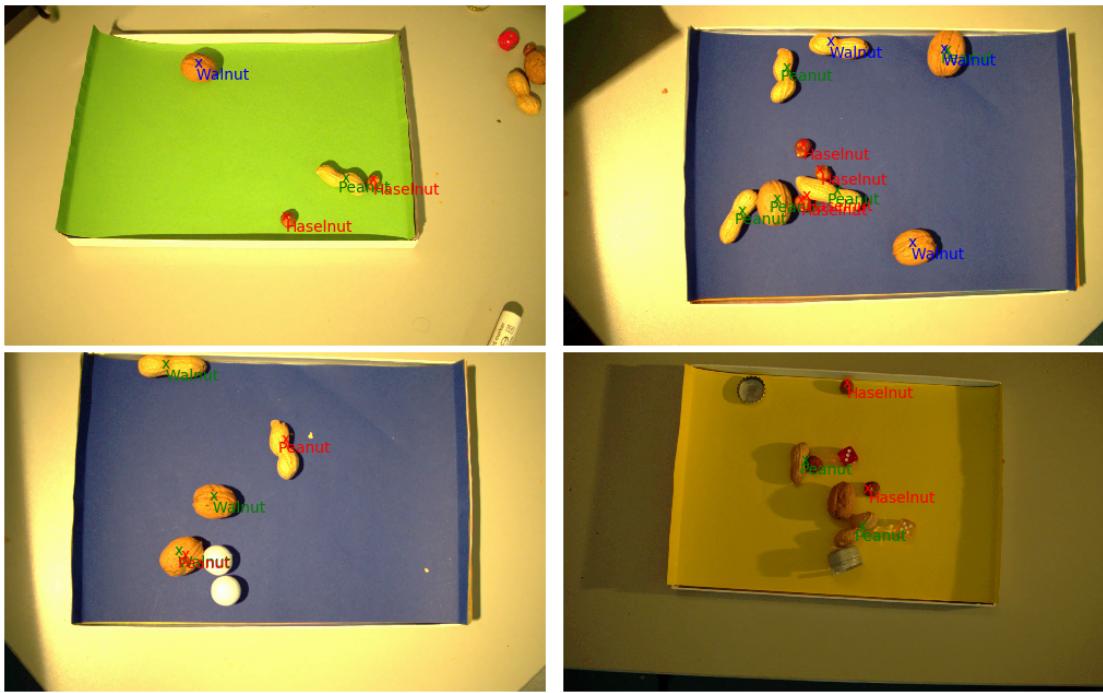


Fig. 8: Results obtained from trained model on test data.

### CONCLUSION

The nut detection system to detect and localize hazelnut, peanut and walnut is developed in this project. Colour histogram is used to detect the stable frame and deep learning framework is used to detect the nuts in input image. The output of the nut detection system is the pixel value and its label is generated in CSV file. The training was performed with 304 images and 76 videos are used to test the approach. The results on the test data shows that our approach is successfully able to classify the nuts but sometimes it is getting confuse in peanut and the walnut.

## REFERENCES

- [1] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, Dec 2001.
- [2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) - Volume 1 - Volume 01*, CVPR 05, page 886893, USA, 2005. IEEE Computer Society.
- [3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [5] Kuo-Yi Huang. Detection and classification of areca nuts with machine vision. *Computers & Mathematics with Applications*, 64(5):739 – 746, 2012. Advanced Technologies in Computer, Consumer and Control.
- [6] Narendra V G, Anita S. Kini, and Anita S. Kini. An intelligent classification model for peanuts varieties by color and texture features. *International Journal of Engineering & Technology*, 7(2.27), 2018.
- [7] Mehdi Abbaszadeh, Aliakbar Rahimifard, Mohammadali Eftekhari, and Hossein Ghayoumi Zadeh et al. Deep learning-based classification of the defective pistachios via deep autoencoder neural networks. *CoRR*, abs/1906.11878, 2019.
- [8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
- [9] CyberAILab. A closer look at yolov3, 2018. <https://www.cyberailab.com/home/a-closer-look-at-yolov3>, Last accessed on 14-01-2020.