Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Edible nut detection in videos using deep learning

Computer Vision

Jan 15 , 2020

Anirudh Narasimamurthy Jayasimha

Introduction

Object Detection : locating instances of objects in the image frame.

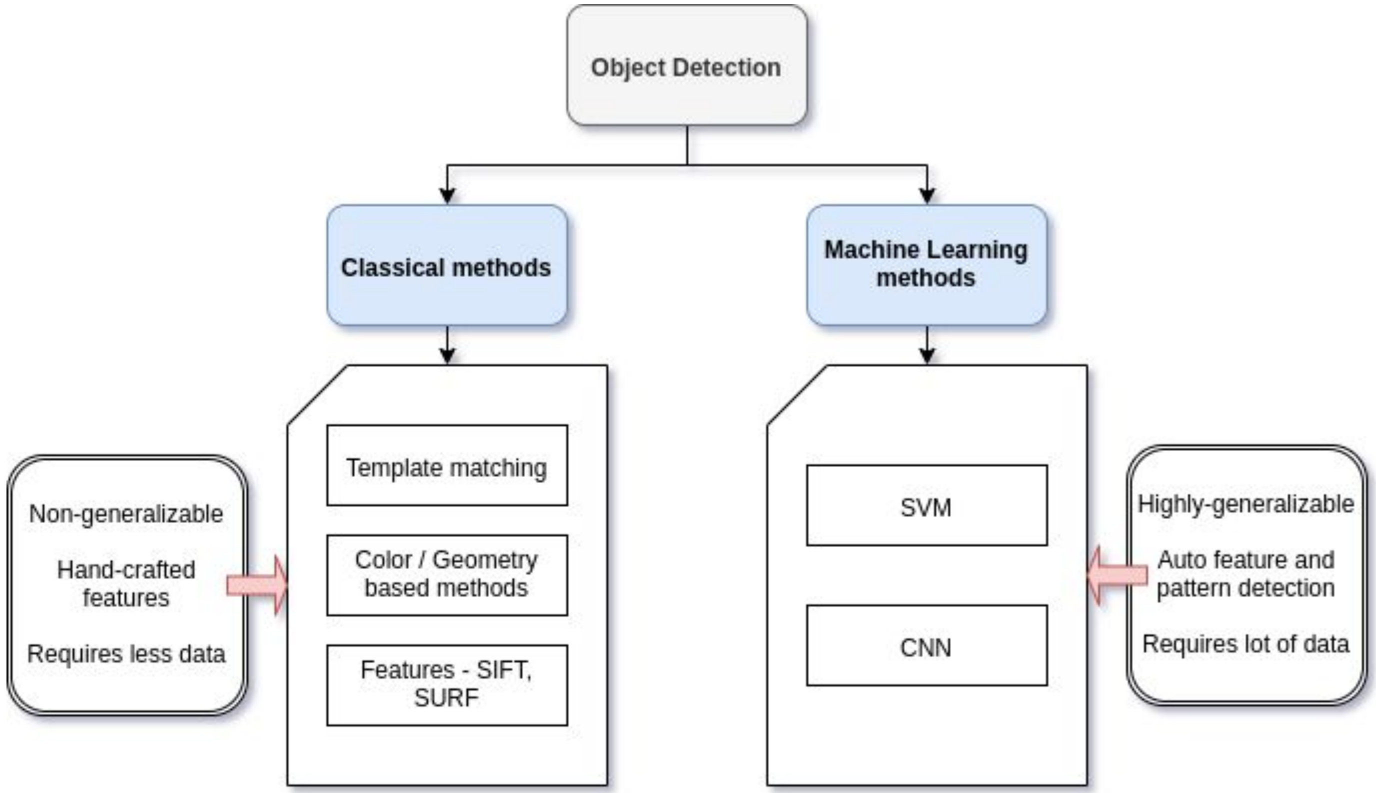


Fig 1. General taxonomy of object detection.[4][7]

Problem statement

- The task of the project is to "*detect of three classes of edible nuts in a video using computer vision*".

{Hazelnut, Peanut, Walnut} ∈ **Nuts**

{Dice, Caps, Pens, Sugar cube, ….} ∈ **Distractors**

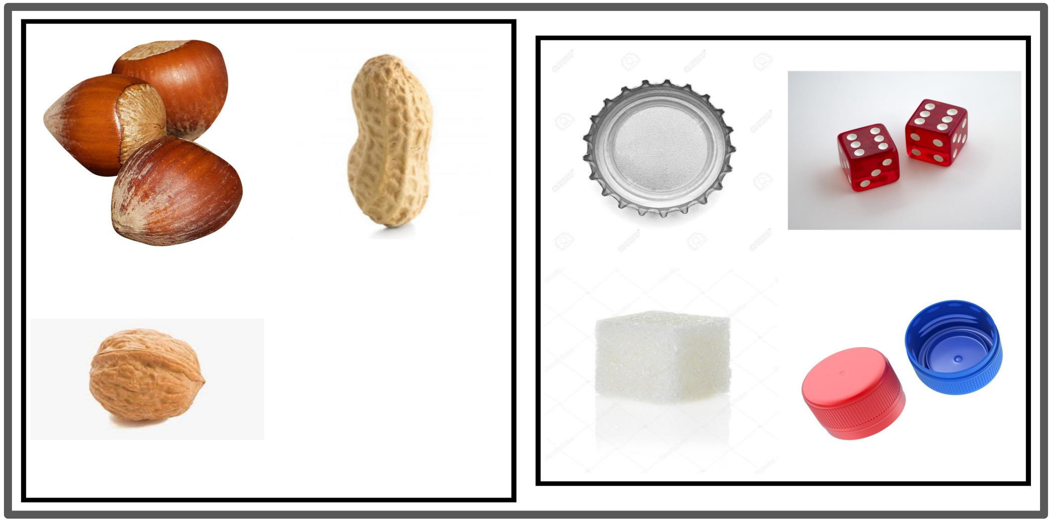


Fig 2. Image showing different classes of edible nuts and distractors

Pipeline

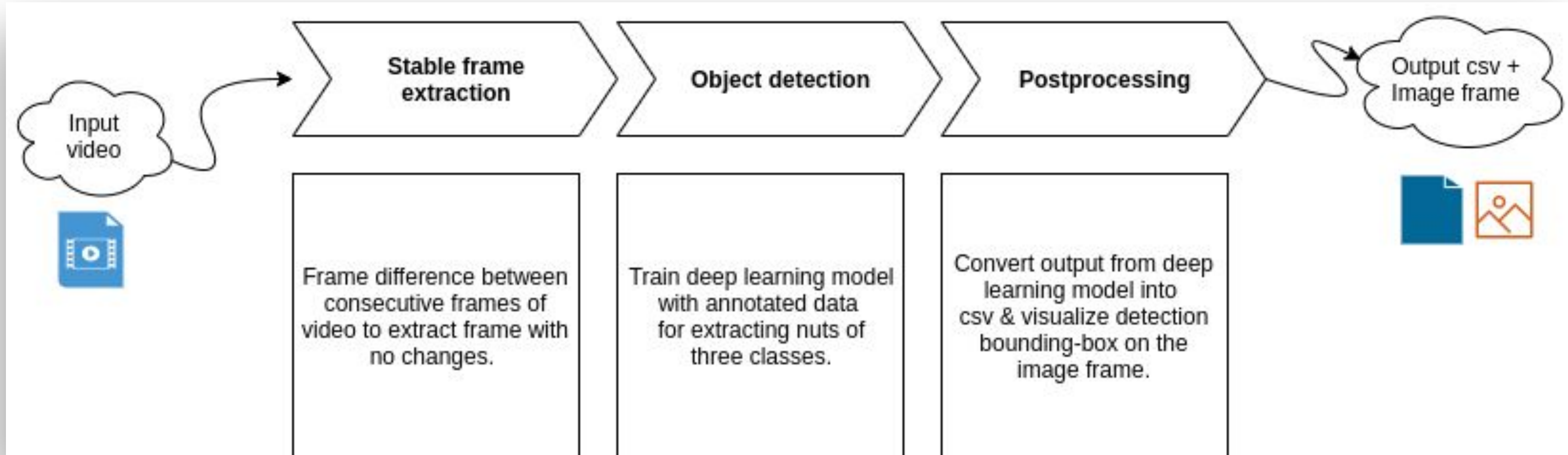
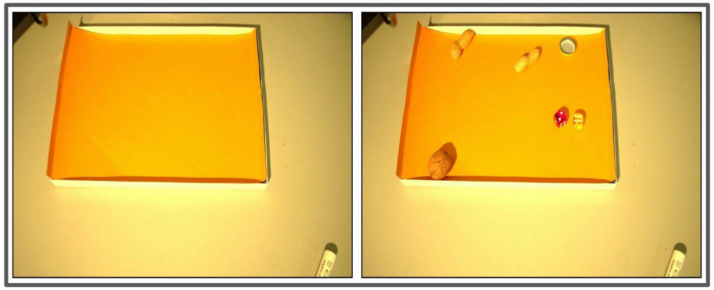


Fig 3. Steps for using deep learning for the task of nuts detection on a video

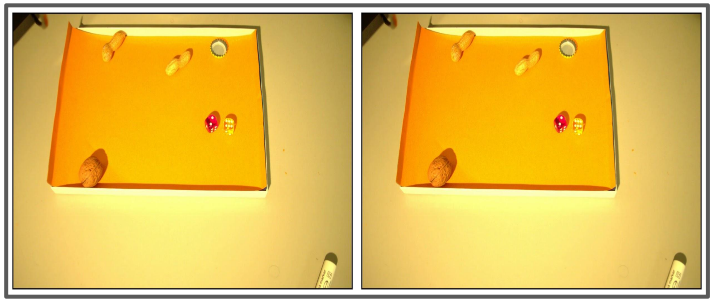
Major consideration when selecting the algorithms were, it should be robust under different under **different lighting conditions**, **high detection accuracy, less false positives, precise localization in frame.**

Stable frame extraction - 1

- **Unsuccessful methods**
  - **Image-hashing** : Technique for fingerprinting images. Eg. p-hash [4] , d-hash etc.
    - Similar images have very close hash values.
    - Dissimilar images have very different hash values.



a) Difference between the hashes is a positive number as the images are very different



b) Difference between the hashes is 0 although the images have a slight difference

Slightly similar images also have the same hash value so they can't be used for extracting stable frame.

Stable frame extraction - 2

- **Successful method :**
  - **Frame-difference :** Take absolute difference between images.

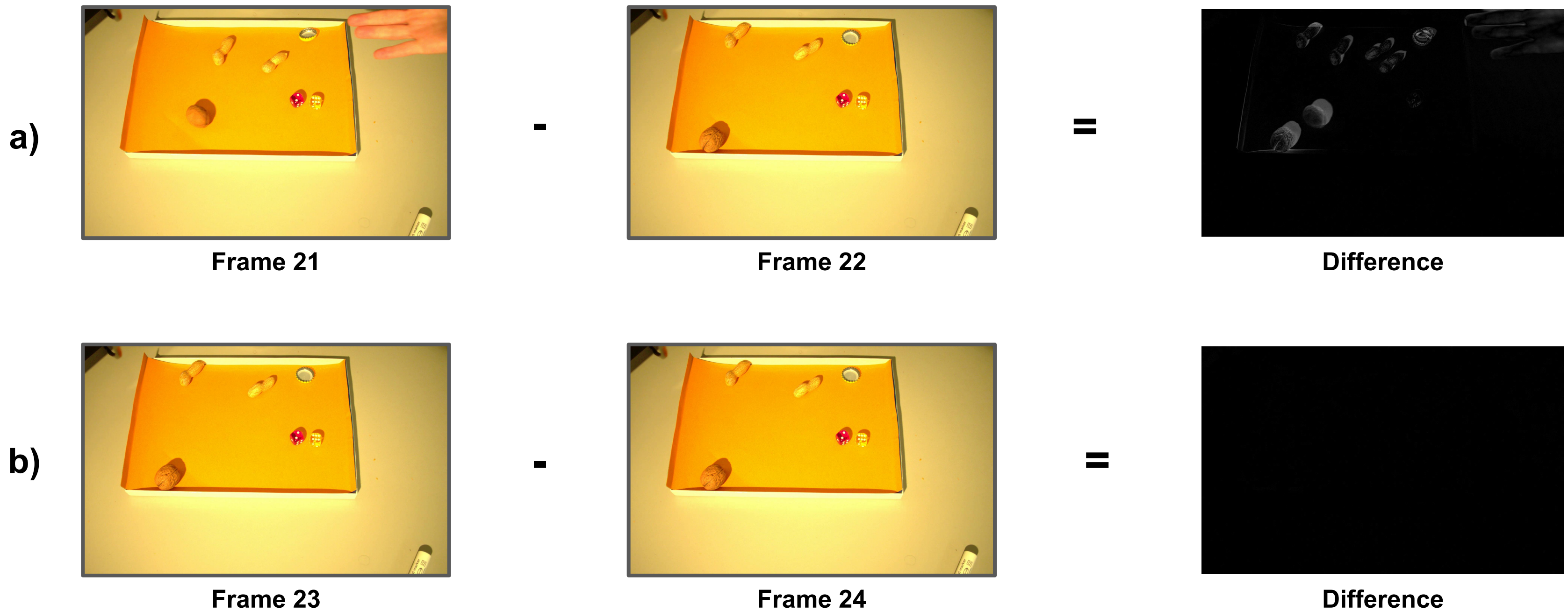


Fig 4. a) The difference between the frames with moving objects leads to an output with high number of non-black pixels.
b) The difference between the frames when there is no change leads to an output with pixels are black.

Stable frame extraction - 3

- Algorithm uses the difference frame extracted to measure the movement between the frames and a threshold is set to return the frame when the movement is below it.

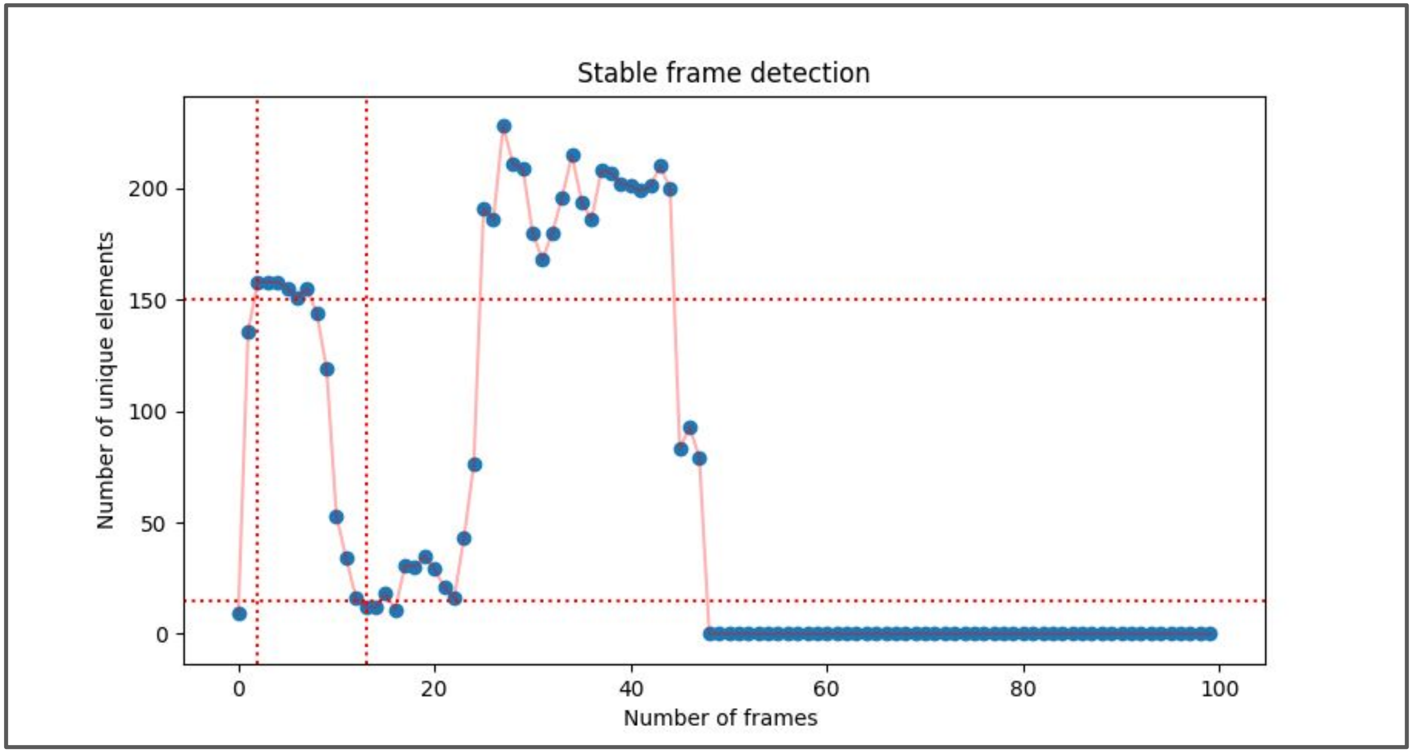
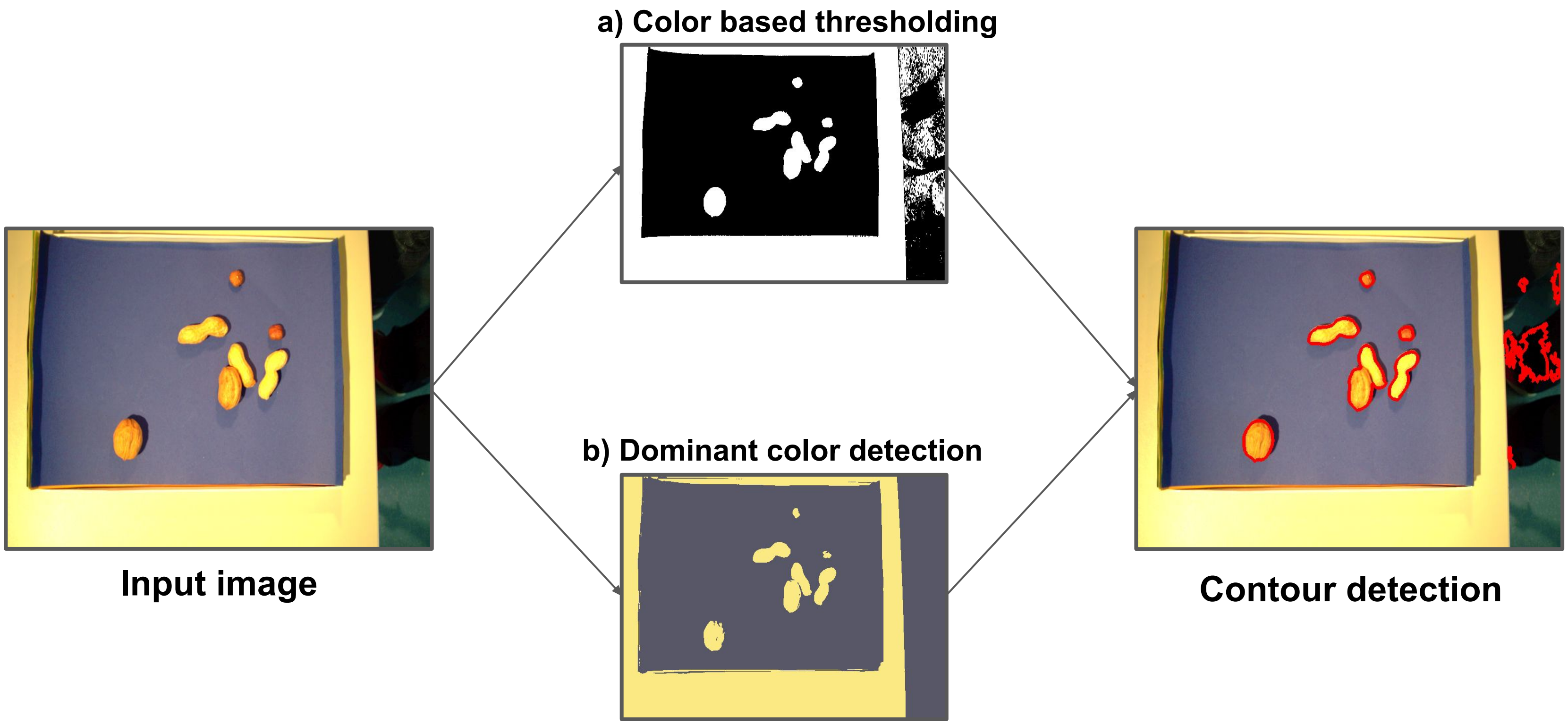


Fig 4. Graph shows the number of unique non-black elements found in the consecutive difference images across the video. The flags are set in the algorithm at start threshold of 150 and stop threshold of 15.

Object detection - 1

- **Unsuccessful methods:**



Joint contours were leading to a failure in obtaining good results.
How do we classify contours ? Colors, Size, Shape?
What about distractors ?

Object detection - 2

- **Successful method :**
  - **Deep Learning :** Train a CNN with annotated dataset

- We use annotated data of 3 classes of nuts and one custom class :
  - Hazelnut
  - Peanut
  - Walnut
  - **Tray (Custom annotated)**

An **FR-CNN**[1] model was trained using our custom annotated dataset using transfer learning on TensorFlow framework[2].

Inference using the FRCNN model done using **OpenCV DNN** [6] module.

Object detection - 3

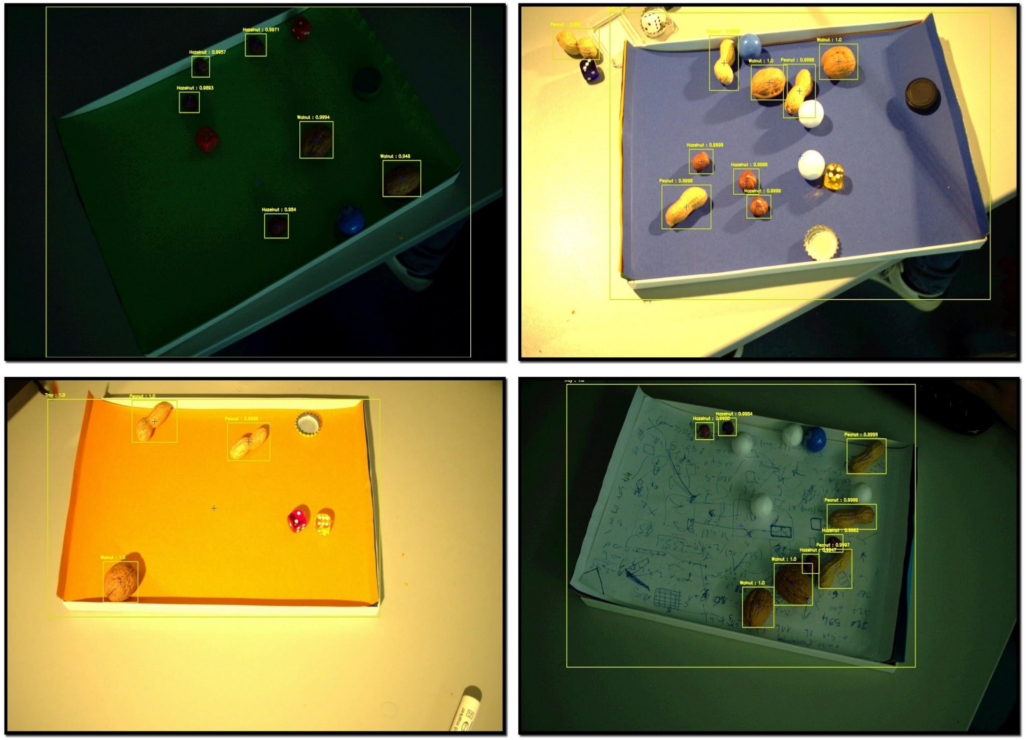


Fig 6. Objects kept under different lighting conditions are also detected.

Deep Learning

Advantages :
1. Works well under diverse lighting conditions.
2. Very less or no false positives.
3. No-hardcoded features (color, shape) are required.

Disadvantages :
1. Large annotated dataset.
2. Size of the model is large.
3. It is a black-box algorithm.

Postprocessing

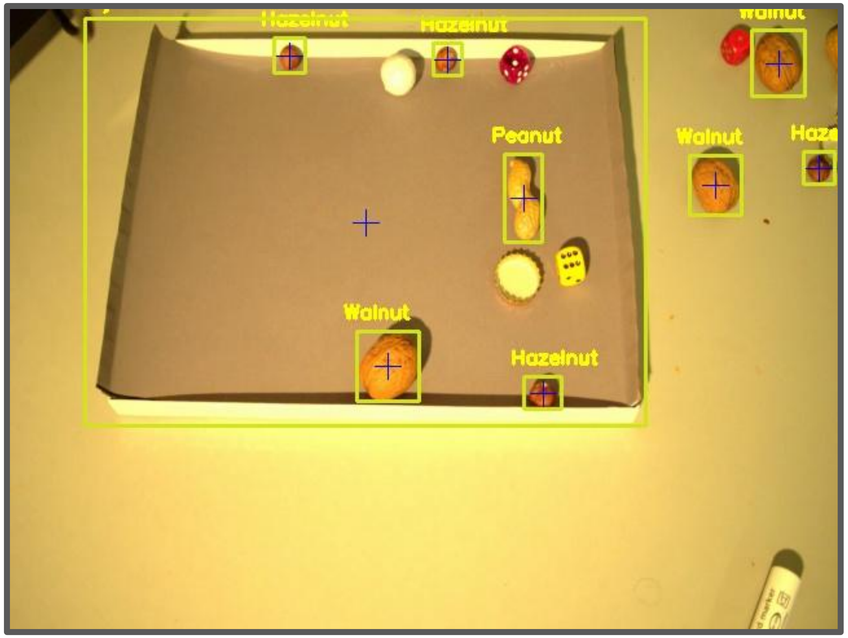


Fig 6. Objects outside the tray are also detected skewing the results. The first step of result post-processing involves removing all the bounding boxes that lie outside the tray.

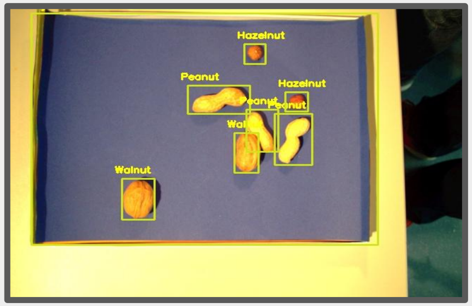
- The extra annotated "**Tray**" class will be used to remove the detected objects that are outside the colored paper on the tray.
- The extracted **bounding boxes centres**, the detected object **class label** along with the **frame number** of the most stable frame are written into as '*.csv*' file.

Expected results

Input Video

Object Detection Pipeline

Result



15, 158, 282, Walnut
15, 317, 206, Peanut
….

Fig 7. This figure shows the input stable frame and the output image and .csv with the bounding boxes and labels after object detection.

Expected failures

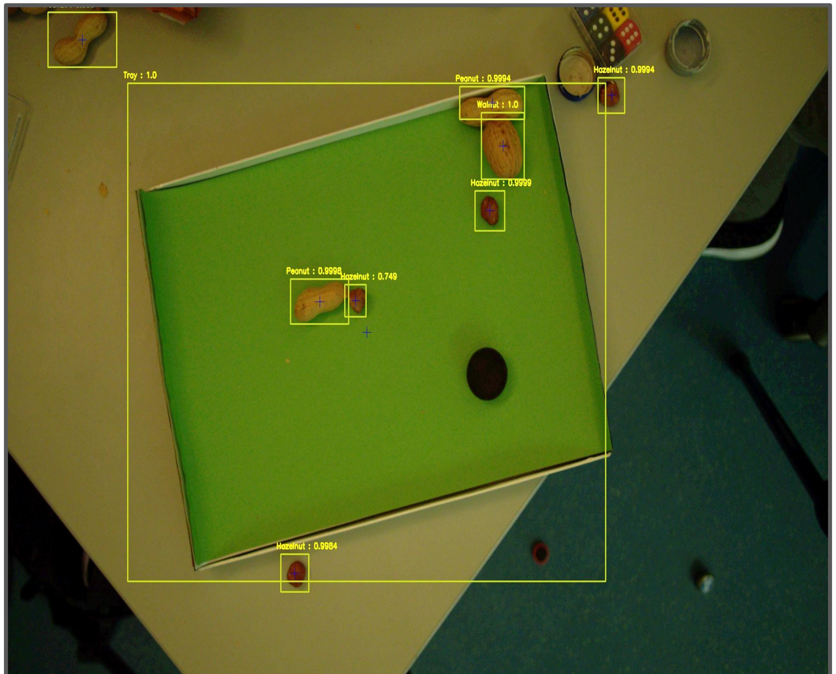


Fig 8.a) Failure in postprocessing when the nuts fall inside the tray bounding box.

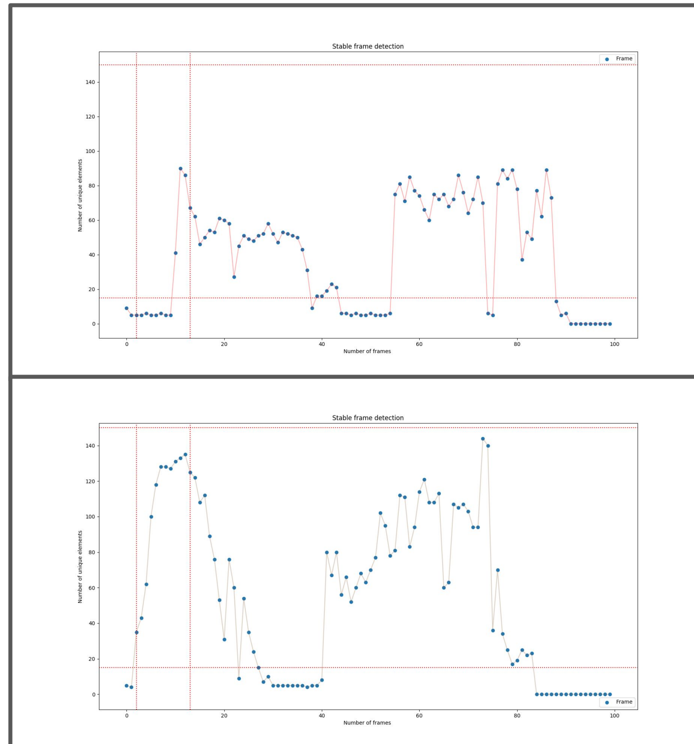


Fig 8.b) Failure in stable frame detection due to wrong thresholding.

References

[1] Ren, Shaoqing, Kaiming He, Ross B. Girshick and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2015): 1137-1149.

[2] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu and Xiaoqiang Zhang. “TensorFlow: A system for large-scale machine learning.” OSDI (2016).

[3] Lowe, David G.. “Object recognition from local scale-invariant features.” Proceedings of the Seventh IEEE International Conference on Computer Vision 2 (1999): 1150-1157 vol.2.

[4] Perceptual hashing, http://phash.org , 12/01/2020.

[5] Zhou, Wengang, Houqiang Li and Qi Tian. “Recent Advance in Content-based Image Retrieval: A Literature Survey.” ArXiv abs/1706.06064 (2017): n. Pag.

[6] OpenCV, https://opencv.org.

[7] Zou, Zhengxia, Zhenwei Shi, Yuhong Guo and Jieping Ye. “Object Detection in 20 Years: A Survey.” ArXiv abs/1905.05055 (2019): n. pag.

Thank You !