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

Dec 11 , 2019

Anirudh Narasimamurthy Jayasimha

Introduction

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

Nuts ∈ {Hazelnut, Peanut, Walnut}

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

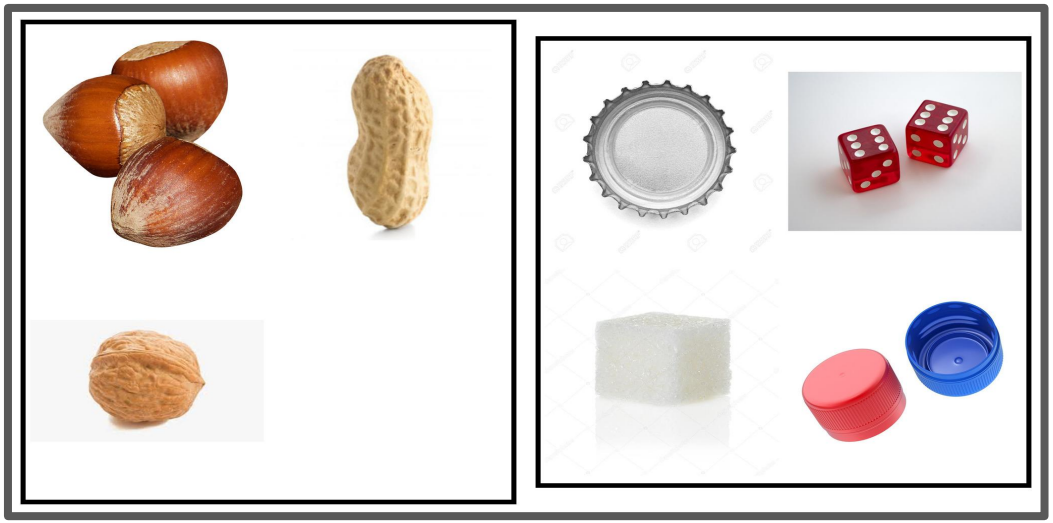

Fig 1. Image showing different classes of edible nuts and distractors

Pipeline

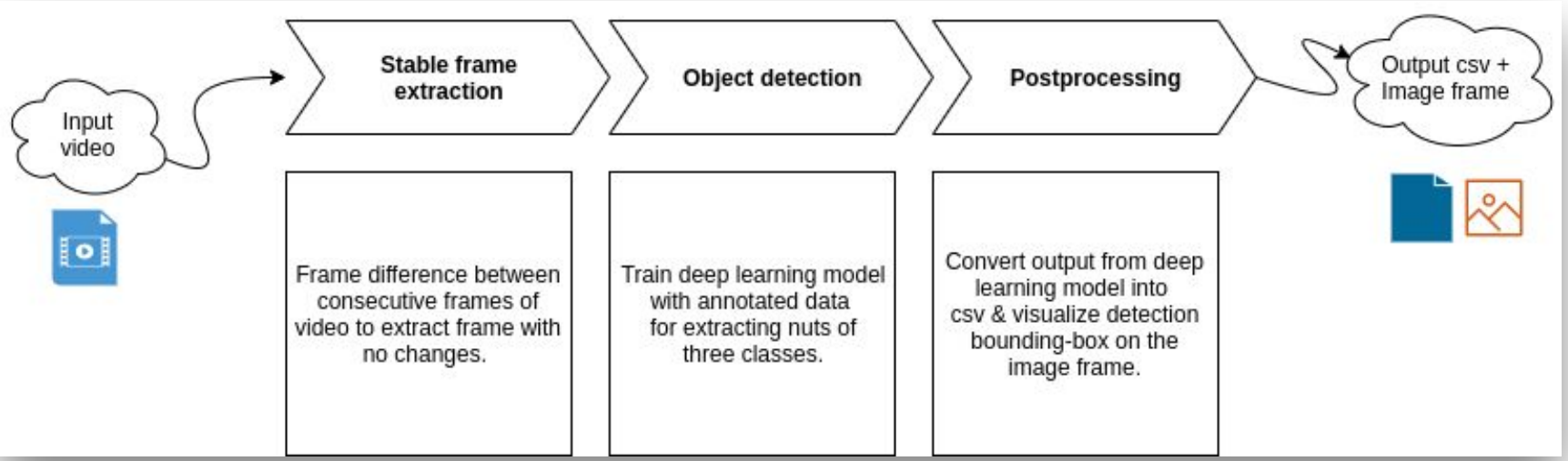

Fig 2. Steps for using deep learning for the task of nuts detection on a video

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

Few other methods that were tested are **flood fill algorithm**, **color based segmentation.**

Stable frame extraction

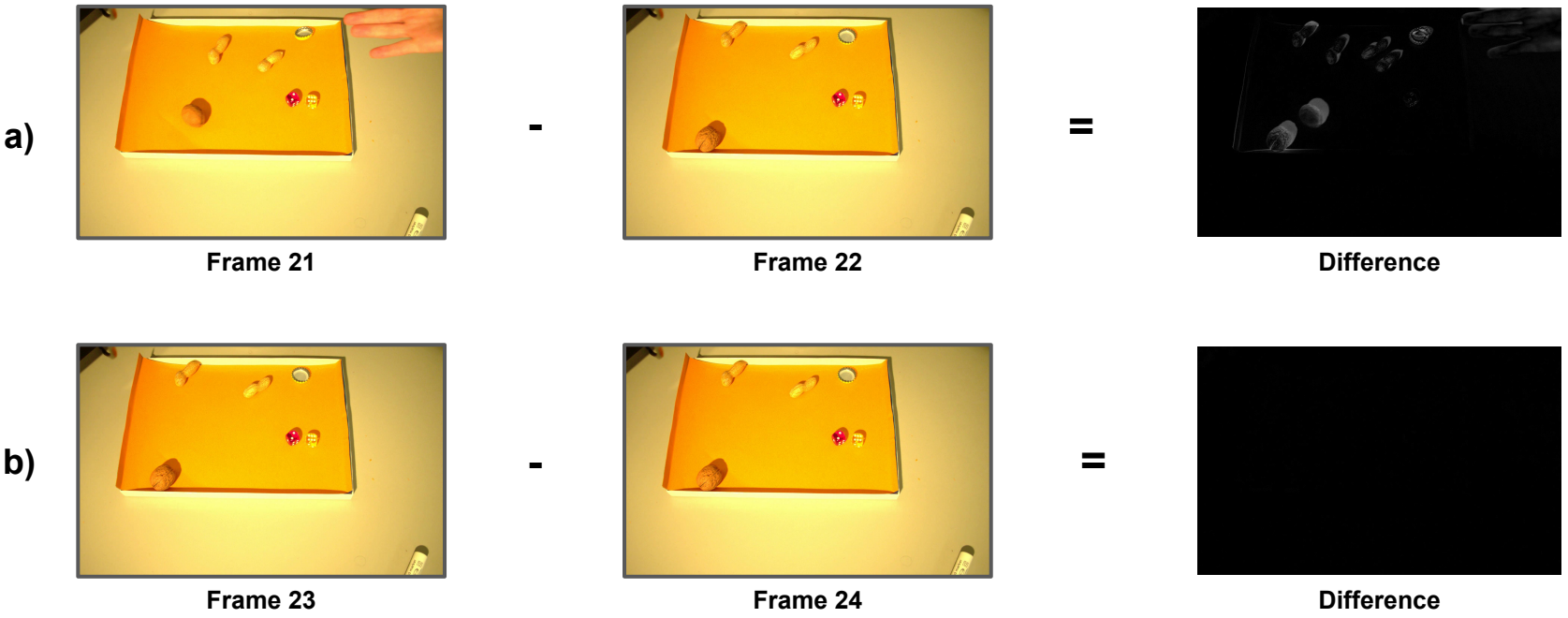

Fig 3. a) The difference between the frames with moving objects leads to an output with high number of non-black pixels.
b) The difference between the frames when there is no change leads to an output with pixels are black.

Object detection

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

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

Inference using the FRCNN model done using **OpenCV DNN** module.

Postprocessing

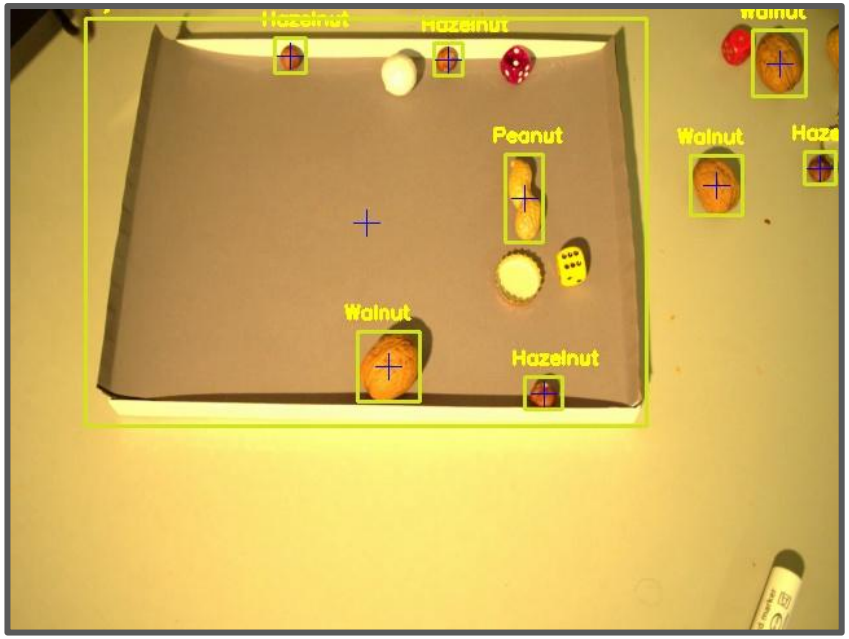

Fig 4. Objects outside the tray are also detected skewing the results. The first step of result post-processing involves removing all the bounding boxes that lie outside the tray.

- 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

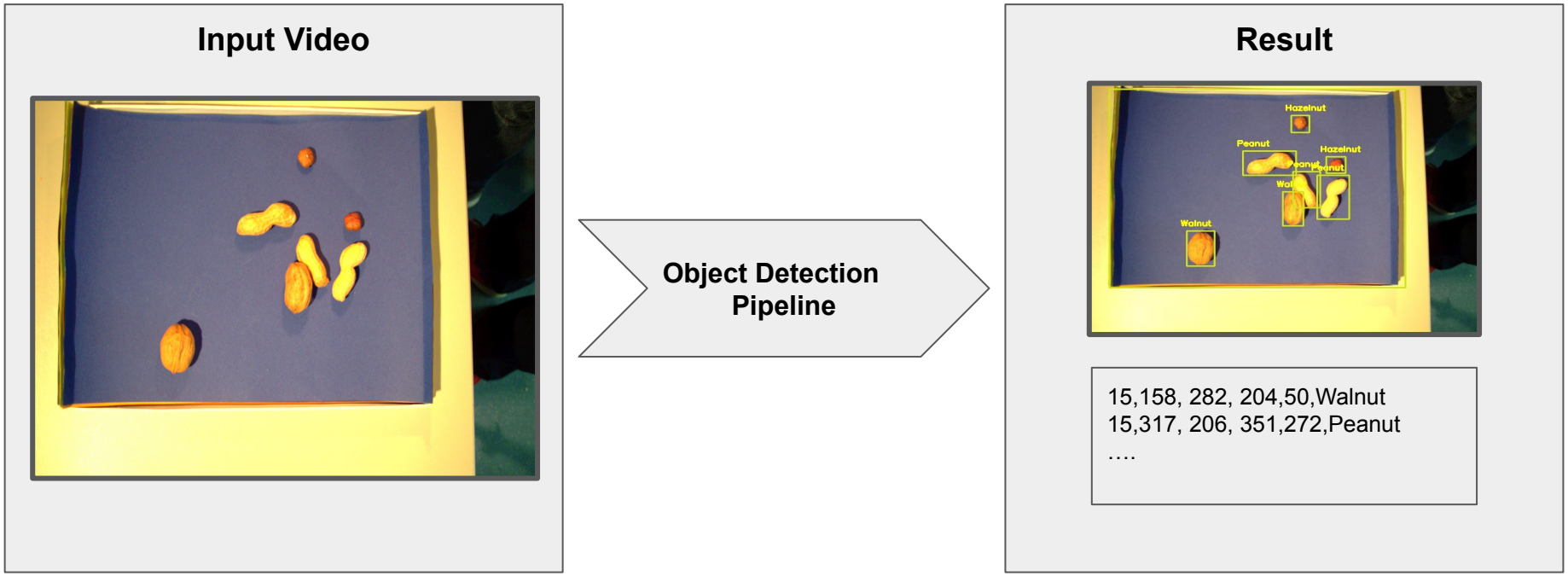

Fig 5. This figure shows the input stable frame and the output image and .csv with the bounding boxes and labels after object detection.

Thank You !