Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Spotting Cap using Deepnet

February 23, 2021

Ganesamanian kolappan-9038581

Introduction and motivation

- Bottle caps are used to seal the bottles [9].
- Bottle caps are widely used in the packaging section in manufacturing industries such as beer, cool drinks, and pharmaceuticals [4].
- Bottle caps are of different types, such as crown cork, flip-top, screw cap, and plastisol [4].
- Object detection made its presence from day to day used smartphones to the autonomous vehicle which is the main motivation for this project [7][13].

Problem statement

- The pivotal aspect of this project is localization, detection and classification of caps namely- faceup, facedown, and deformed.
- The detection should be robust across different variations such as
  - Illumination
  - Viewpoints
  - Frame rates
  - Background
  - Region Of Interest (ROI)
  - Different distractors like peanut shell, pens, coins, toys etc.

  The challenging distractor is coin since the cap and coin are of identical shape. Additionally surface tends to be reflective.

State-of-the-art

- Traditional detectors: Viola Jones detector, Scale Invariant Feature Transform (SIFT) [12], Histogram of Oriented Gradients (HOG) [6], and Deformable Part-based Model (DPM) [13].
- Deep architectures: One-stage [8], Two-stage [5] and Keypoint joint learning [2].

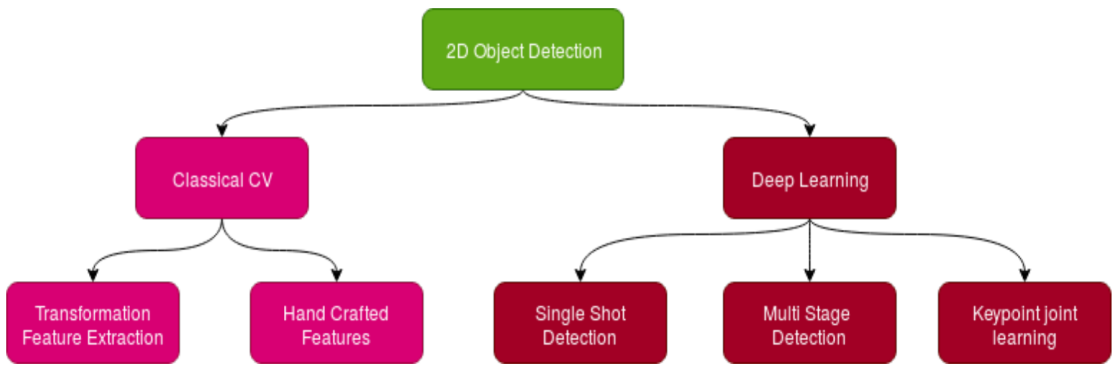


Figure 1: *2D object detectors [11].*

Difficulties faced for classical method

(a) No ROI

(b) Caps and coins

Figure 2: *Difficulties encountered for classical method.*

- Image (a) shows, dataset from one particular sample doesn't abide the procedure of rectangular boundary.
- Image (b) depicts, coins and facedown cap tends to be similar due to factors such as background, illumination, object's reflectance.

Proposed strategy

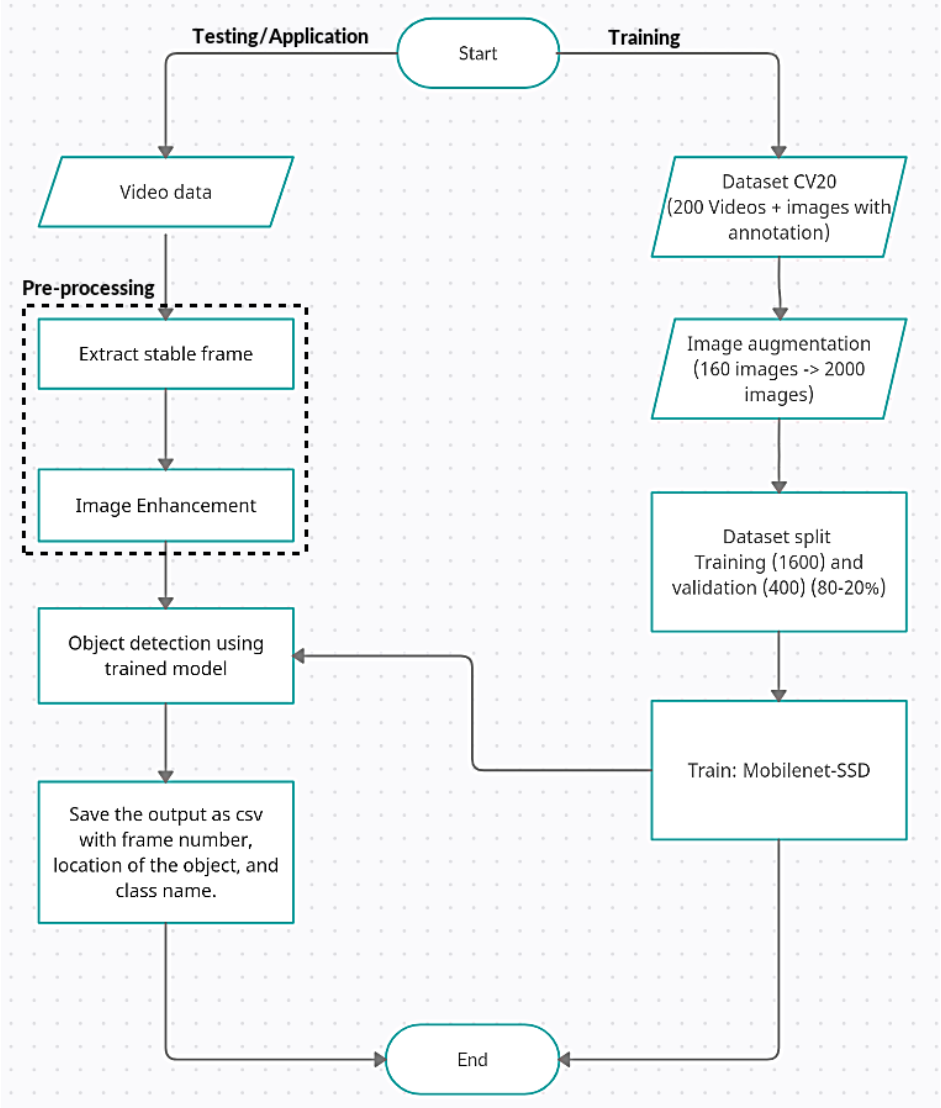


Figure 3: *Workflow of this work.*

Assumptions

- Objects in the stable frame are static.
- Occlusions are valid.
- Object outside the ROI is ignored.

Material used

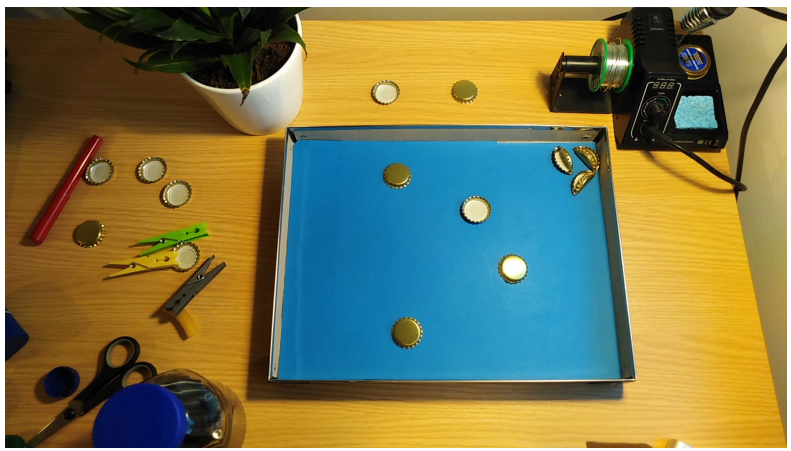
- Python $>=$ 3.5.
- OpenCV $>$ 3.4.1.
- Tensorflow = 1.15.
- LabelMe = 4.1.1.
- Numpy $>=$ 1.16.
- Pandas $>=$ 0.25.
- Scikit learn $>=$ 0.22.
- Platform for Scientific Computing at Bonn-Rhein-Sieg University [3].
- Dataset: CV20 video_package 1 and 2.

Training phase

- Random sampling 160 images from 200 images for training.
- Augmenting the selected 160 images to 2000 images.
- Splitting 2000 images in 80-20% for training and validation, respectively.
- Running the SSD trained model [11] on the new images to detect bottle caps.
- Saving the model for application/testing.

Image augmentation

- In many deep learning applications, data augmentation is momentous to teach the network to be robust across various input objects.
- Augmentation is performed using default Tesnorflow functions on the original images such as random crop, random saturation, horizontal and vertical flip.



(a) Provided image

(b) Augmented Image

Figure 4: *Augmented Image.*

Single Shot Detector (SSD)

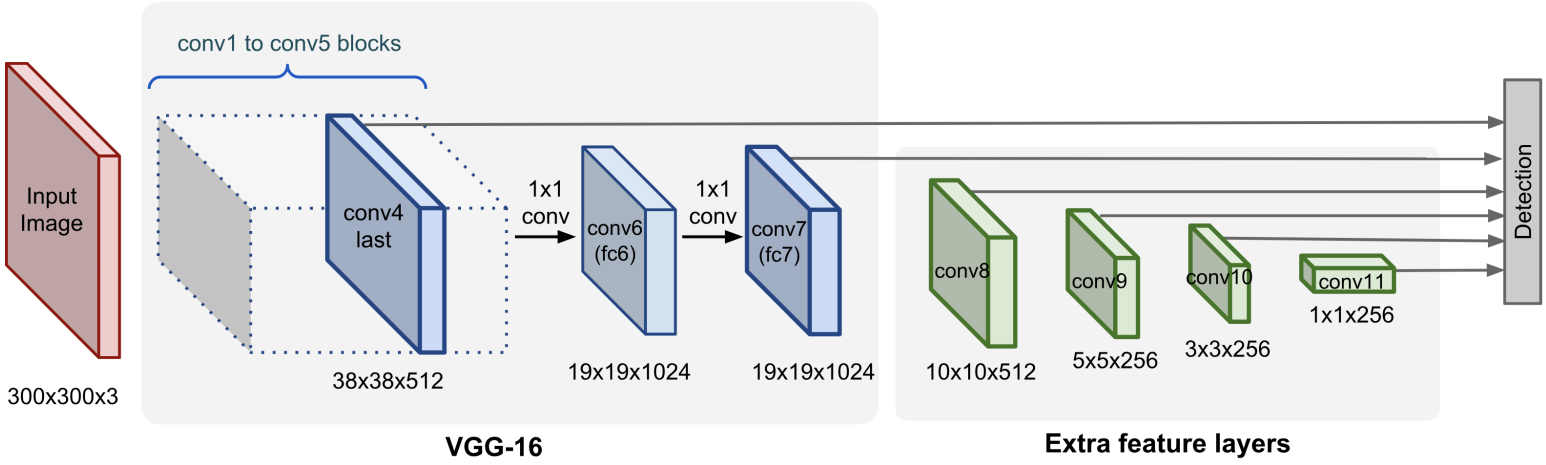


Figure 5: *SSD architecture [5].*

VGG-16 is used to extract the feature map, additional layers are needed for video analytics. In case of image data fully connected layer can be used after VGG-16.

Trained Single Shot Detector (SSD)

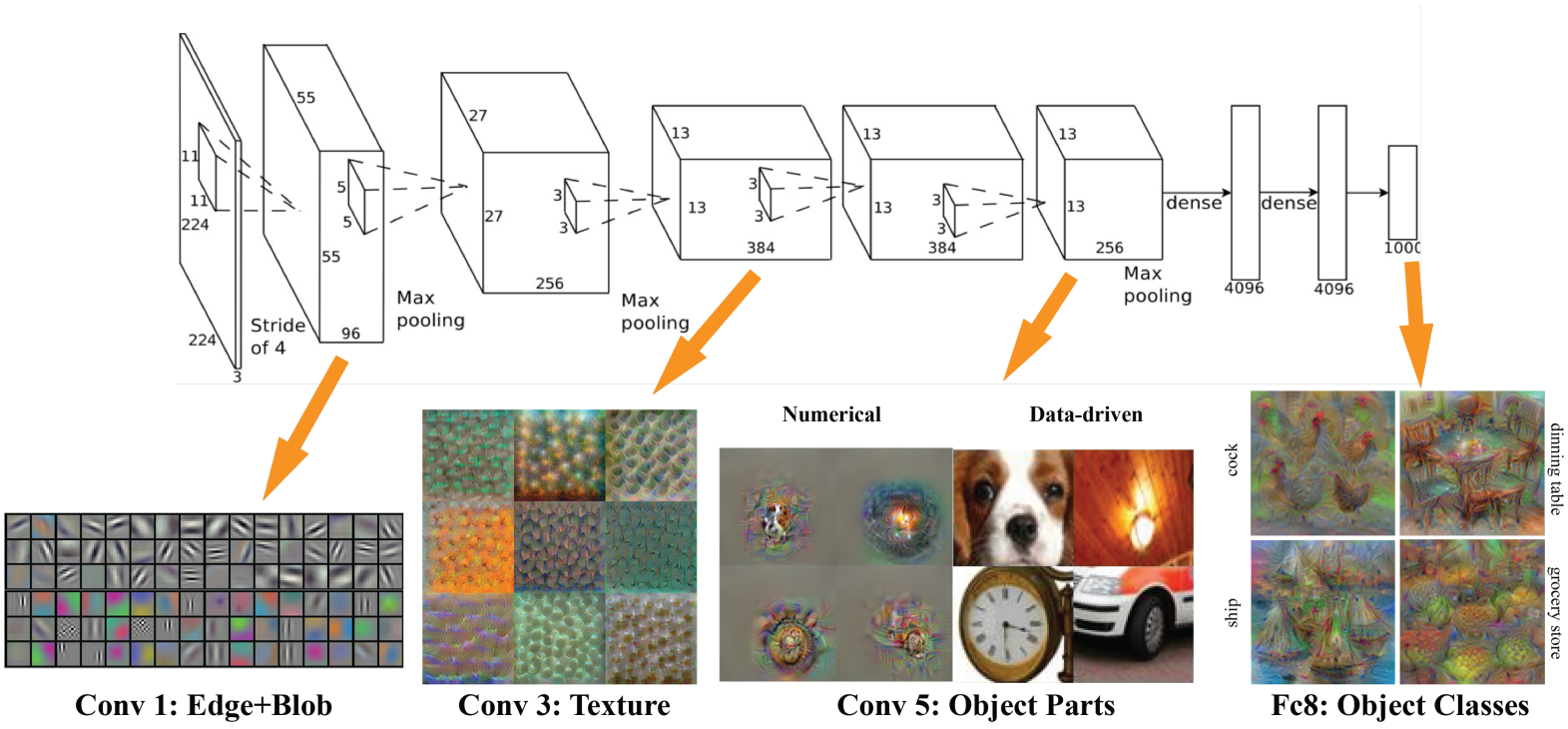


Figure 6: *Trained SSD feature map on the coco dataset [5].*

The output of VGG-16 can able to detect various objects learned from the training set.

Single Shot Detector (SSD)- anchor box

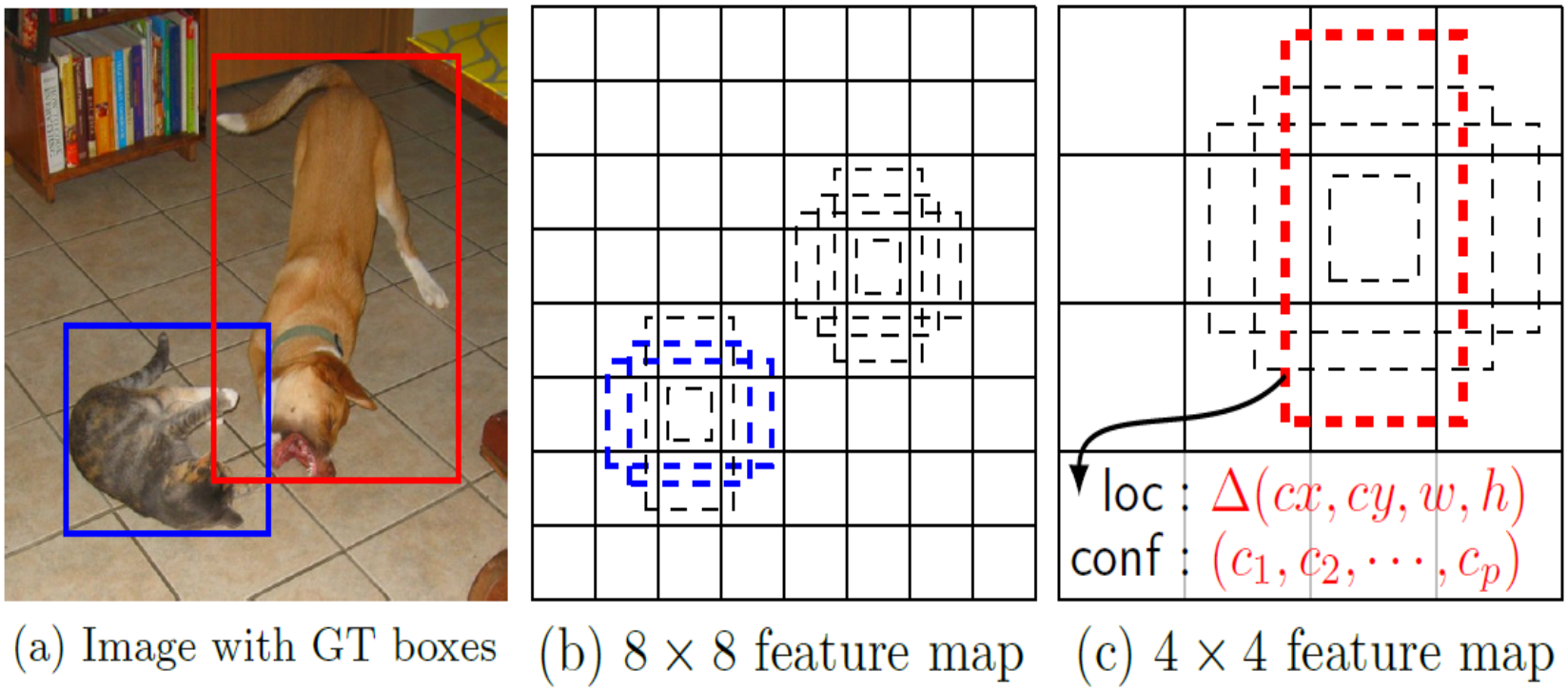


Figure 7: *SSD- Multiple anchor boxes [10].*

Hyperparameters for Single Shot Detector (SSD)

Hyperparameters	**Value**
Optimizer	Adam
Initial learning rate	0.001
Weight initialization	Xaviar
Batch size	32
Epochs	2000
Weight decay	0.0005
Learning rate decay factor	0.94

Table 1: *Hyperparameters for SSD.*

Performance of Single Shot Detector (SSD)

The loss value reaches the least of 0.08. Increasing the epochs or changing the batch size does not show any improvements.

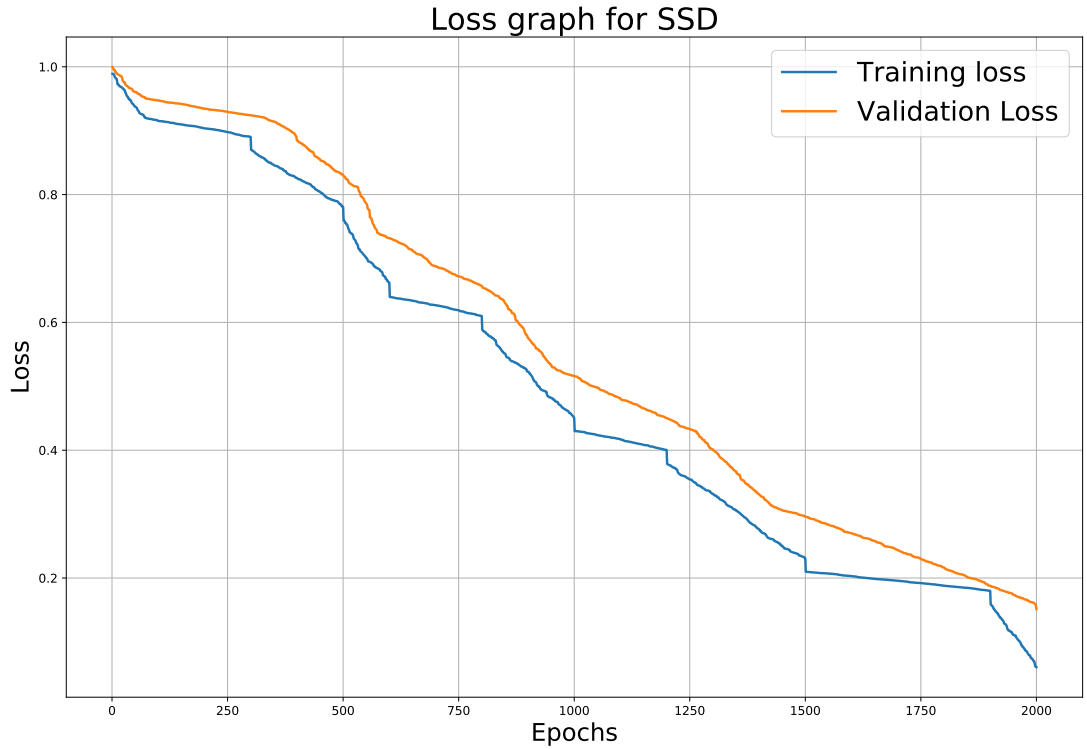


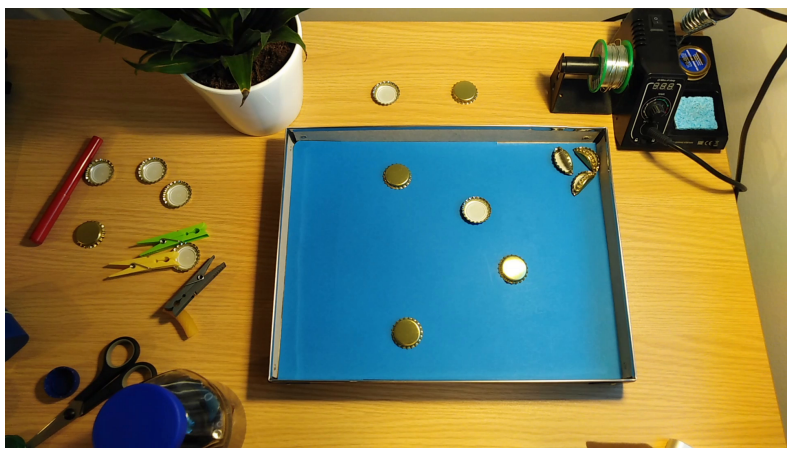
Figure 8: Metric loss of SSD on the training images of 1600 and validation images of 400.

Application/ testing phase

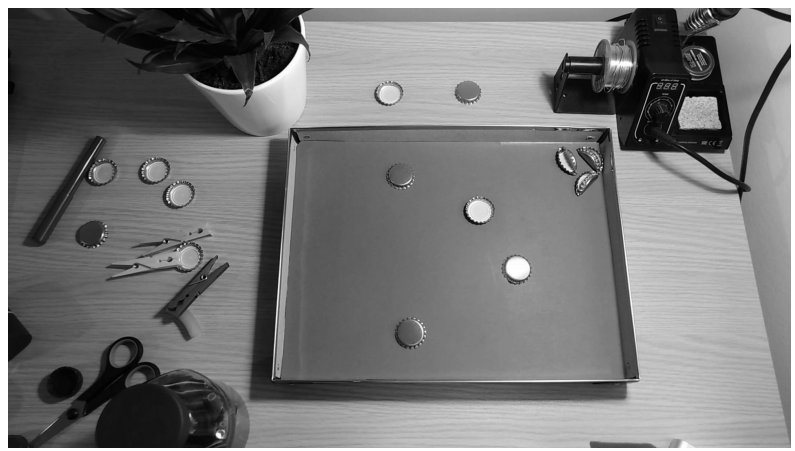
- The remaining 40 images of the source video is provided as input.
- Frame extraction from the video input where the objects are stable.
- Image enhancement to compensate the over exposed and under exposed image
- The above two steps are called pre-processing.
- Trained model is executed on the pre-processed image to detect, localize and classify the bottle caps.
- The output is stored in a csv format as each detection in a single line with the stable frame number, co-ordinates of the cap followed by the class type.

Frame extraction

- The frame extraction is performed by the frame differencing, after the Gaussian blur and canny edge.
- Initial few frames and last few frames are ignored to concentrate on the center static frames with bottle caps.



(a) Provided image



(b) Extracted image

Figure 9: *Stable frame extraction the first image (a) is the reference image as provided. However, image (b) is extracted by the proposed method.*

Efficiency of frame extraction

Among 40 videos, the stable frame is extracted aptly for 36 videos.

$$\text{Efficiency} = \frac{\text{Number of videos with correctly extracted frame}}{\text{Number of testing videos}} \text{X} 100 \quad (1)$$

Therefore, Efficiency of Frame Extraction $= \frac{36}{40} * 100 = 90\%$.

Failure situations of frame extraction

The frame extraction fails when the objects are dealt fast and when there is a higher frame rate (greater than 25).

(a) Objects are dealt fast

(b) Higher frame rate - 30fps

Figure 10: *Failure in stable frame extraction.*

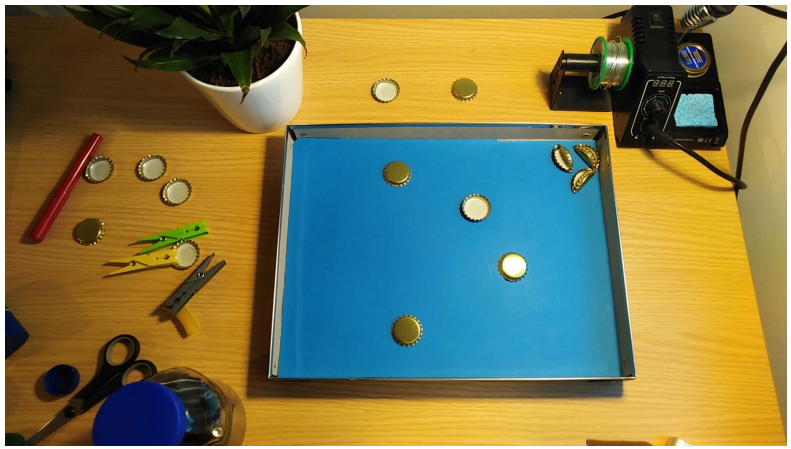
Image enhancement (1/2)

- Image is enhanced by a method called Contrast Limited Adaptive Histogram Equalization (CLAHE) [14].
- Procedure is adapted from [1].

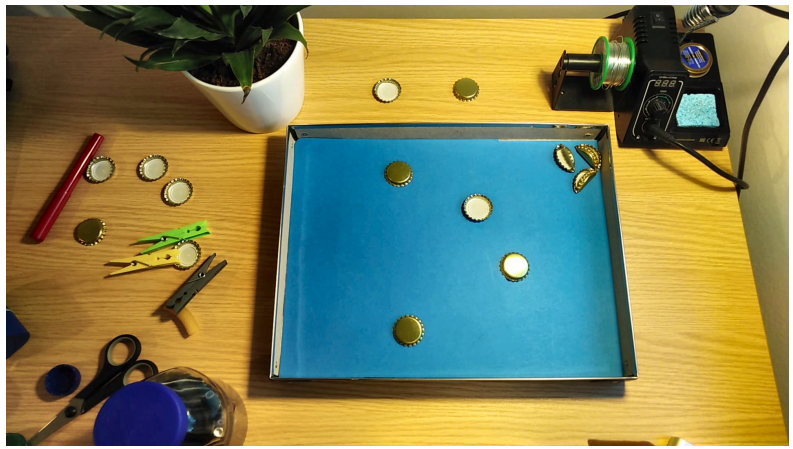
(a) Provided image **(b)** Enhanced Image

Figure 11: *Image enhancement for under exposed image, the first image (a) is the reference image as provided. However, image (b) is an underexposed which is extracted and enhanced by the proposed method.*

Image enhancement (2/2)



(a) Provided image



(b) Enhanced Image

Figure 12: *Image enhancement for slightly over exposed image, the first image (a) is the reference image as provided. However, image (b) is an slightly over exposed image extracted and enhanced by the proposed method.*

Evaluation of the method - localization

Correction difference = Number of caps located correctly - Number of cap located incorrectly

$$\text{Localization accuracy} = \frac{\text{Correction difference}}{\text{Total number of caps}} \text{X} 100 \qquad (2)$$

The total number of caps in the 36 videos is 201, number of caps located correctly is 187, whereas number of caps classified wrongly is 26. Therefore,

$$\text{Localization accuracy} = \frac{187 - 26}{201} * 100 = \frac{161}{201} = 80.1\%$$

.

Evaluation of the method - classification

The classification of bottlecaps is evaluated by F1-score.

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

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

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Where, TP is True Positive, TN is True Negative, FN is False Negative, and FP is False Positive.

Classification result - confusion matrix

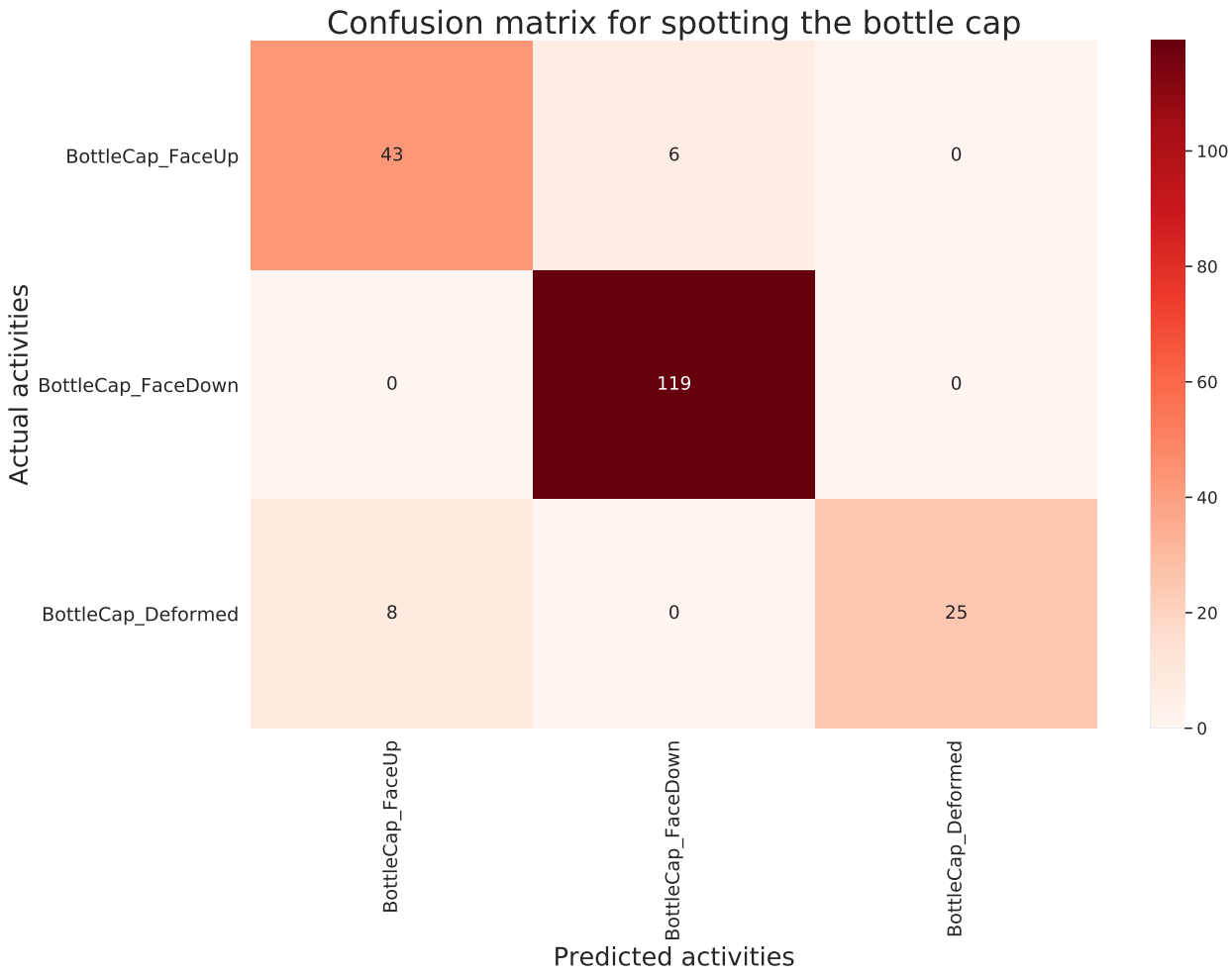


Figure 13: *Confusion matrix for spotting the cap.*

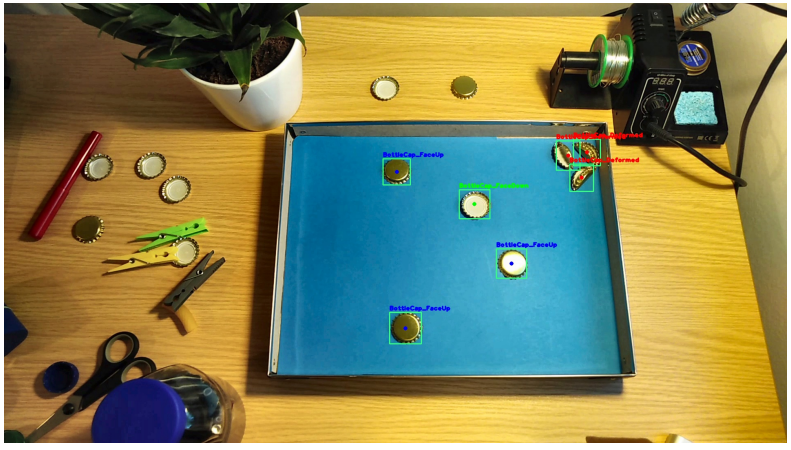
Classification result - classification report

Metric	Faceup	Facedown	Deformed
Precision	0.84	0.95	1
Recall	0.87	1	0.75
F1-score	0.85	0.97	0.85

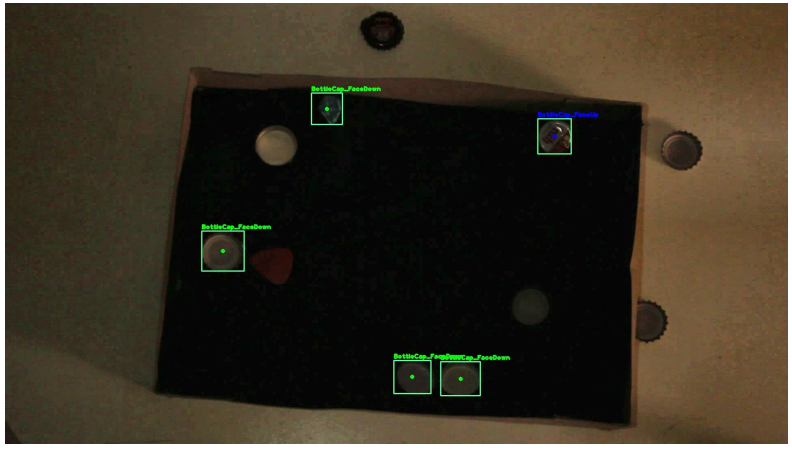
Table 2: *F1-score for each classes.*

Mean F1-score = $\frac{1}{3}\sum_{i=1}^{3} F_i = \frac{0.85+0.97+0.85}{3}$ = 0.89

Detection result



(a) Slightly over exposed image



(b) Under exposed image

Figure 14: *Bottlecap detection using MobileNet-SSD.*

Future work

- Robust frame extraction.
- Data augmentation concentrated on a single object.
- Should compare the performance of YOLO and classical method.

Reference (1/5)

Dibya Bora, Anil Gupta, and Fayaz Khan.

Comparing the performance of l*a*b* and hsv color spaces with respect to color image segmentation.

06 2015.

Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Košecká.

End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching.

02 2018.

Hochschule Bonn-Rhein-Sieg (H-BRS).

Platform for Scientific Computing at Bonn-Rhein-Sieg University, 2020.

Accessed on: 2020-12-21. [Online].

Reference (2/5)

R. Kulkarni, S. Kulkarni, S. Dabhane, N. Lele, and R. S. Paswan.

An automated computer vision based system for bottle cap fitting inspection.

In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–5, 2019.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg.

SSD: single shot multibox detector.

CoRR, abs/1512.02325, 2015.

David G. Lowe.

Distinctive image features from scale-invariant keypoints.

International Journal of Computer Vision, 60(2):91–110, 2004.

Reference (3/5)

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.

You Only Look Once: Unified, Real-Time Object Detection.

arXiv e-prints, page arXiv:1506.02640, Jun 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

arXiv e-prints, page arXiv:1506.01497, Jun 2015.

X. Ren, J. Wen, Y. Lan, T. Li, and X. Wang.

Design of bottle cap detection system based on image processing.

In *2020 Chinese Control And Decision Conference (CCDC)*, pages 4880–4885, 2020.

Reference (4/5)

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen.

MobileNetV2: Inverted Residuals and Linear Bottlenecks.

arXiv e-prints, page arXiv:1801.04381, Jan 2018.

Dat Tran.

Raccoon Detector Dataset, 2021.

Accessed on: 2021-02-21. [Online].

Zhengxia Zou, Z. Shi, Yuhong Guo, and Jieping Ye.

Object detection in 20 years: A survey.

ArXiv, abs/1905.05055, 2019.

Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye.

Object Detection in 20 Years: A Survey.

arXiv e-prints, page arXiv:1905.05055, May 2019.

Reference (5/5)

Karel Zuiderveld.

Graphics gems iv.

chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

Thank you