Internship report on

“An implementation of AI based conveyer belt object detection and counting at Food and Beverage Industries.”

Submitted by:

Mohammad Raghib Noor

ID No.029-012-128

MIS 12th Batch (BBA)

Department of Management Information Systems

Faculty of Business Studies

University of Dhaka

Submitted to:

Md Fahami Ahsan Mazmum

Assistant Professor

Department of Management Information Systems

Faculty of Business Studies

University of Dhaka

Letter of Submission

17th November, 2021

Md Fahami Ahsan Mazmum

Assistant Professor

Dept. of Management Information Systems

Faculty of business studies, University of Dhaka

Subject: Submission of internship report **on "An implementation of AI based conveyer belt object detection and counting at Food and Beverage Industries".**

Dear Sir,

I have completed my internship in Essential Infotech **(from April 25th, 2021 to July 25th, 2021)** and would like to submit my internship report as per your specifications. I would also like to draw your kind attention to the fact that I have tried my level best to gather and organize all the information needed for this particular report, and in doing so, I have tried my utmost to live up to your standards. The report has been prepared under your competent supervision and I respectfully acknowledge your guidance. I put my best effort here and hope it will able to meet your expectations. Thank you for your heartiest co-operation.

Sincerely yours,

Mohammad Raghib Noor

MIS 12th Batch (MBA)

ID: 029-012-128

12.128.noor@gmail.com

Department of Management Information Systems, University of Dhaka.

Declaration

I do hereby declare that the internship report on **“An implementation of AI based conveyer belt object detection and counting at Food and Beverages Industries”** has been prepared by me for the partial fulfillment of BBA program from the **Department of Management Information Systems (MIS), Faculty of Business Studies, University of Dhaka**.

I further affirm that the work reported in this internship report is original and it has not been submitted by any other students for the completion of BBA or any other degree.

…………………………………

Mohammad Raghib Noor
ID: 029-012-128
MIS 12th Batch
12.128.noor@gmail.com
Department of Management Information Systems
University of Dhaka.

Supervisor’s certification

This is to certify that, Mohammad Raghib Noor, ID: 029-012-128, student of Department of MIS, University of Dhaka, has completed his Internship Program **entitled “An implementation of AI based conveyer belt object detection and counting at Food and Beverages Industries”**. He is allowed to submit this report as the partial fulfillment of BBA degree.

He has done his job according to my supervision and guidance with his best effort to do this successfully. I think his dedication to his study will be helpful in the future to build up his career. I wish him every success in life.

…………………………………

Md Fahami Ahsan Mazmum

Assistant Professor
Department of Management information Systems
University of Dhaka.

Acknowledgement

First and foremost, I would like to say none of what I have done in this report would have been possible if it were not for my loving, kind and benevolent creator, Allah (SWT)'s will. All praise be to Him. Secondly, right from the top, I would like to mention my father. The pioneer of the mathematician inside me, my biggest fan and my best friend. My father spends his day tirelessly working and struggling for the family, but not a night goes by when he sees me working on something Im passionate about, and he doesn't skip his talk shows to sit with me and watch me work. Even in my LinkedIn profile, most of my project updates include my father and me performing demos of the new updates in my projects. Thirdly, I would like to mention my mother, for never complaining about my absence at home as I chase my passions. She tirelessly supported me both mentally and physically with her unconditional love, warm food and words of encouragement. Fourthly, I thank my Sheikh Zillani sir for being there as a pillar of support for me when very few believed in me. It would belittle my gratitude towards him if I were to describe it in words. I also acknowledge the encouragement and assistance given by all the people who were involved both directly and indirectly in the preparation of the project and this report and apologize to the people whose names are not mentioned in this paragraph but their contribution is highly valued by me.

My next gratitude goes obviously to my academic supervisor assistant professor **Md Fahami Ahsan Mazmum sir** for his untiring guidance, help, effort, and suggestion. Without his direct guidance, this report could not be possible. His forbearing personality constantly inspired me to make the report improved. I shall remain ever grateful to him. I would also like to thank my visionary of a CEO at Animo.Ai -sister concern of Essential Infotech, Ridoy Ahmed Sir and Head of Management, Mustafizur Rahman Khan Sir for their limitless patience and guidance in helping me and my team build our project.

Executive Summary

This report is prepared as part of my completion of BBA program where the details and testing results of my internship project is given.

My internship, unlike most BBA graduates, was based in a field that is rigorously based on programming, development and study of tangible products based on AI technology. Thus my report speaks of the project that I was tasked to build in order to automate the inventory tallying from the conveyer belt at a certain client’s biscuit factory (Food and Beverage Industries).

First part of this report is based on the introduction which includes topics introduction, objective of the project, methodology of the project, limitation that I have faced while preparing the project.

The main part includes details on the components and structure of the project and the testing results received from our project’s Proof of Concept. There is details regarding the research development of these components. I have also provided a drive link to an hour long video that provides visual proof of this project’s accuracy at counting items (boxes and cartons) in a conveyer belt with varying difficulties of orientation. [A2]

At the end of the report, some recommendation, conclusion, reference are given which would be helpful for the proper understanding of the report.

In a brief, this report contains all the essential information related to the development of a computer vision object counter. I had a wonderful experience in building the project and writing my internship report.

Table of Contents

Chapter-1
Introduction

1.1 Rationale of the Study

Many a times in large industries, when a firm grows too big through the help of its economies of scale and growth, it faces diseconomies of scale and starts to face losses which often times might even spell the company's demise. A crucial reason for this is a lack of communication and control between the central authority and the individual process centers. In most production lines, a crucial place for corruption or displacement of products would be at the conveyer belts, where most of the tally counts are done manually by shifts of human labor, thus there is not much security and guarantee in the data provided by them as the data can be easily manipulated, given the hierarchical and physical distance between central authority and lower level employees in large industry settings. In order to bring absolute assurance in the tally counts of products passing through the conveyer belt, the best bet would be to eliminate the space for human error with computer vision and AI technology. Hence, the project and study developed and conducted would not only help automate the counting procedures in firms but it also eliminates the chances of corruption and prevents displacement of products within the production line. The counting done also provides reference to cross check on inventory counts against real time counts of goods produced in production lines.

1.2 Objective of the Study

1.2.1 Broad objective

The broad objective of the study is to provide details on a computer vision-based camera counter prototype that can help control and monitor the flow and count of inventory of boxes of products at different industries and saves employee wages for the company by automating the tallying requirements with technologies based on computer vision and database management.

1.2.2 Specific Objective

This study mainly aims to:

1. Provide details on the structure of an AI based computer vision object counter
2. Provide findings found in training the AI model, writing the programming script and testing the product.
3. Provide reference to video evidence of testing accuracy of the AI project.

1.3 Methodology

The report is technical and descriptive in nature. It provides descriptions along with references to researches of technologies used in the internship project, along with information on development and test findings and insights of the project along with their numerical evaluation. The research design and methods of this study are described as follows:

1.3.1 Data Collection

This study is based on my internship project's development and live test findings and insights. The necessary data for this study were collected through expert knowledge, programming documentations, research paper study and live testing of internship project. Thus there are both primary and secondary sources of data.

1.3.2 Data Sources

The "primary sources" of data and information include-

- Personal observation.
- Model training and testing observations

The "secondary sources" of data and information include-

- Research Papers based on computer vision technology
- Programming documentations
- Various, articles, compilations etc. regarding python programming language, object-oriented programming and object detection and tracking.

1.4 Organization of the report:

This internship report is mainly organized through three major parts:

1.4.1 Basic Information on the project

Introduction, literature review, organization description is involved in this part. Introduction part is separated into diverse segments like rationale of the study, objectives, research methodology etc. This part describes the objectives of the project this report is based on, the theories and research involved in the technologies used in this project. An overview of the firm under which I have made this project, "Essential Infotech", is included.

1.4.2 Analysis and Findings of the report

Data Analysis of Training data and their findings are involved in this part. All the related data analyzed for the project are represented in the data analysis section. Results and empirical data collected from the analysis are included and discussed about in findings of the study.

1.4.3 Conclusion and Recommendation

The Conclusion section consists of conclusive statements related to evidence from data analysis along with visible computational bottlenecks of the system and its spaces for improvements. The recommendation section consists of suggestions of various degrees of preference that maybe used to fix the bottleneck of the system and thus help the system achieve higher degrees of efficiency and output.

1.5 Limitations of this project:

While training the model used, I had to face a number of problems that hampered the smooth development of this project and this prolonged our testing and training period. These limitations were:

- Lack of quantity video data.
- Lack of quality assurance in video data collection.
- Lack of quality in video data.
- Administrations at testing venue may have many protocols to follow before allowing access to industrial video data.
- Lack of data access in industries.
- Extensive data security provides problems for flow of training and testing.
- Internet speed provided in Bangladesh may provide difficulty during passage of data from backend to frontend user interface and this may cause disruption in the counter project's application.
- Increasing conveyer belt camera line connections would require very high GPU usage

Chapter- 2
An Overview of Essential Infotech

2.1 Background of the organization

Essential Infotech’s AI sector had its inception in the March of 2020. Ever since its inception the sector was called Animo.Ai. This sector of the company develops and provides web-based ERP solutions to businesses along with tailored business intelligence software. Essential Infotech is situated in Uttara, with a growing, technically sound and specialized workforce ideal for handling software requirements of industries and firms. One of the areas of specialization of Essential Infotech’s Animo.Ai is IOT development and AI based IOT solutions. According to the vision of the organization, the company shall go on about to being one of the first movers in domestically producing industry grade IOT based AI technologies in Bangladesh.

2.2 Essential Infotech Vision

The vision of Essential Infotech is to build and provide an industrial grade AI based ERP product that applies most of the senses a human being applies to his or her surrounding- Sight, Hearing and Touch. A product that can analyze and provide automated reports of data from every corner of an industry with these senses stimulated by IOT sensors and personal cloud systems.

2.3 Essential Infotech Mission

- To provide high quality ERP software products in a B2B format.
- To research on AI technologies
- To outsource cheaper built AI technologies and scripts.
- To create a B2B AI industry through provision of its services to firms and industries.

2.4 Essential Infotech Motto

Committed to providing AI solutions to any kind of business problem and automation of routine industrial work.

Chapter-3
Literature Review

The AI counter project is basically consisting of two main components and technologies in use:

- Object Detection
- Video Tracking

3.1 Object Detection

Object Detection is the processing of images to find specific objects in the images given certain inputs to an algorithm. Object Detection has been a part of computer vision for a very long time, even as early as 2005.

The first proper human pattern detector was made by:

(Dalal and Triggs; 2005) [1]. This detector was based on The Sliding Windows Detection technology, the paper says that a window is serially passed from one area of an input image to the next until the whole image is analyzed by that window, and this way, the area where the object in the window is detected.

The method of feature extraction used in order to detect the images is based on evaluation of standardized and mean subtracted local histograms of image gradient orientations within a dense grid. The algorithm considers the distribution of image intensity gradients and edge directions and emphasis within the sliding window in order to detect a segment of the image to say whether there is a human being or not. The HOG detector has been evaluated against other feature extractor-based detectors such as Haar wavelets, PCA-SIFT and shape context approaches. Results showed that the HOG-based detectors were far better than the performance provided by the Haar wavelets, PCA-SIFT as well as the shape context approaches, giving almost impeccable separation in the MIT test.

After many years of using the sliding windows technology, a more automated and computationally inexpensive technology was developed in 2016. This technology revolutionized Object Detection for good. It was known as **You Only Look Once: Unified Real-time Object Detection,** a paper

[1] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, 20-25

written by **Joseph Redmon, S. Divvala, Ross B. Girshick, Ali Farhadi.** It provides a method to unify both the technology of image classification as well as finding out where in the image the classified object exists (localization/detection) with nothing but the use of a 106 layered Convolutional Neural Network. This one paper has even been further researched upon and updated in the form of acronyms **YOLO** (Joseph Redmon Et Al., 2016), **YOLO 9000** (Joseph Redmon Et Al; 2017), **YOLO V3** (Joseph Redmon Et Al; 2018**), YOLO V4** (Alexey Bochkovskiy Et Al; 2020), **YOLOV5** (Xingkui Zhu Et Al; 2021)

For our project, we used YOLO V3 as our object detector as it was very accurate and we did not have the computational resource to train a YOLO V4 or V5.

YOLO V3 (Joseph Redmon Et Al; 2018) [2]: YOLO V3 is a 105 layered convolutional neural network, compared to its predecessors the YOLO V3 comes along with anchor layers of 3 sizes that is **13 x 13**, **26 x 26** and **52 x 52 from an input of 416 x 416 dimensioned image**. This means that the image is detected on 3 scales for better accuracy. The output is of a matrix of equal length and width but of 5 + n channels where n is the number of classes. **This means that the image is segmented within the algorithm into say, 14 x 14 pieces, then each of those pieces would consist of 5 + n layers**, where the **first layers** represents **a binary classification** of whether there is an image or not. The **second layer** would consist the **x coordinate of the middle point of the object in decimals**. The **third layer** would contain the **y coordinates of the middle point of the object detector**. The **fourth and fifth layers** would contain the **height and width respectively** of the bounding box that would be used to represent the object detected. The **rest of the layers** represent **a binary vector that denotes which class the object belongs to**. Once all the possible bounding boxes are created the YOLO algorithm uses Generalized Intersection over Union in order to find out the most probable bounding boxes among the clusters of bounding boxes created and then Non Max Suppression is used to eliminate bounding boxes that don't pass a certain confidence threshold. This way, the tasks of both object detection and image classification are accomplished at the same time within the confines of one algorithm.

[2] Redmon, J. and Farhadi, A. (2018) YOLO v3: An Incremental Improvement. arxiv:1804.02767v1 [cs.CV], Unpublished.

Further researches lead to the development of:

YOLO V4 (Alexey Bochkovskiy Et Al; 2020) [3]**:**

The research advancement of the YOLO V4 was based on lowering the GPU demands by its predecessors and running the model in real-time with the use of a conventional GPU. The basic aim of this resource was fast operating speeds for neural networks, in production systems and optimization for parallel computations. This model is basically an amalgamation of models based upon three different neural network architectures The paper proposes backbone layer building based on the likes of CSPResNeXt50 and CSPDarknet53 for GPU usage and uses grouped-convolutional models such as MixNet, GhostNet/MobileNetV3 for VPU usage.

As for the neck layer, the model uses SPP-Net, a convolutional neural architecture that uses spatial pyramid pooling to eliminate the static sized constraints within images. As for the head, the model uses its forefather, The YOLO V3 which we have already discussed about above.

YOLOV5 (Xingkui Zhu Et Al; 2021) [4]**:**

This paper is based on single stage detection on drone captured images. Drone-captured scenarios are of wide sue in plant protection, wildlife protection and urban surveillance. The issues faced in drone-based image detection is that the object scale changes greatly due to changes in flight altitude. Drone based images contain objects of high density, which causes occlusion between objects. Also, drone-based images contain many confusing geographic elements because of large area coverage. In order to tackle these problems, the YOLOV5 is designed with a backbone consisting of CSPDarknet53 and path aggregation networks (PANet) with a neck of TPH-YOLOV5 and 4 detection heads separately used for the detection of tiny, small, medium, large objects. These heads are then replaced with Transformer Prediction Heads (TPH). In order to find regions of attention in images with large coverage, Convolutional Block Attention Modules (CBAM) are applied to sequentially iterate the attention map along channel-wise and spatial-wise

[3] YOLOv4: Optimal Speed and Accuracy of Object Detection; Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao; https://arxiv.org/abs/2004.10934

[4] Yolov5-based channel pruning is used for real-time detection of construction workers' safety helmets and anti-slip shoes in informationalized construction sites; September 2021, Journal of Physics Conference Series 2031(1):012027; DOI:10.1088/1742-6596/2031/1/012027

dimensions. For data augmentation, Photometric distortions and geometric distortions are immensely used by researchers.

3.2 Video Tracking (Danelljan et. al.; 2014) [5]

Video tracking is based on the processing of a certain feature (bounding box coordinates or segmentation coordinates) within each frame in a series of frames within the video, and thus predicts bounding box coordinates for convolutional models that are too computationally expensive or large to run on each frame. There have been many video trackers that have been used so far such as the CSRT Tracker, MOSSE Tracker, Kalman Filters.

In the case of our project, we implemented **correlation tracking based on the paper by Martin Danelljan** in our tracker object class as there was **available documentation** that was relatively easier to apply rather than CSRT or AI based algorithms, given our **limited computational resources during development**.

Danelljan's tracker has been one of the most celebrated tracking algorithms due to its accuracy compared to other the then state-of-the art tracking technologies. The tracker, according to the paper, is based on learning discriminative correlation filters. He states in his paper that, his tracker even though started off as one dimensional, the research was soon extended towards multidimensional applications such as visual tracking, object detection and object alignment. It basically utilizes an n-dimensional mapping of variables in order to represent a signal. The filter is then learned using HOG features. The actual paper also includes many details on experimental setups for transmission of empirical data.

[5] [**Danelljan et al**., BMVC **2014**, PAMI 2017]: https://conferences.mpiinf.mpg.de/dagm/2018/template/GCPR_tracking_tutorial_181009_pdf_ver.pdfPowerPoint-presentation (mpg.de)

Chapter - 4
Description of Internship Job

Before I applied for this job, I was required to intensively brush up on Linear Algebra, Multivariate Calculus as well as learn machine learning techniques and programming syntax. I was primarily recruited as an AI developer intern at Essential Infotech and then shifted to their sister concern Animo. Ai. My internship was non paid with an evaluation period spanning up to 3 months. My job duties included that of a regular employee at the office such as creating weekly logs of my work and submitting it to the HR department, signing in and out of our HRM system to keep track of entry and exit times from office compound, visiting my supervisor first things first in the morning in order to get an overview of my current tasks to be done, etc.

Apart from my regular duties, my duties as an AI developer included:

1) Intensively reading research papers relevant to the technologies needed to fulfill the application demand of our current clientele.
2) Building neural networks or using frameworks provided by research papers in order to test models on their applicability to the needs of our clientele.
3) Creating, cleaning and augmenting train and test image data for training and evaluating image based neural networks.
4) Scripting models in higher level programming languages like Keras.
5) Writing scripts of codes required to create machine-learning pipelines for clientele projects.
6) Leading junior interns in completion of projects.
7) Visiting clientele industry compounds to deploy tested in-house models for further testing.
8) Designing prototype models for clientele Proof of Concept projects.
9) Advising management and finance team on GPU hardware requirements for projects.

This report of mine describes the project that my team and I designed and coded for a clientele factory's tallying needs, thus the whole report would probably describe my job the best.

Chapter - 5
Details on Hardware Testing Resources and Operational Pipeline of the Computer Vision Based Counter

5.1 Overview of Topic

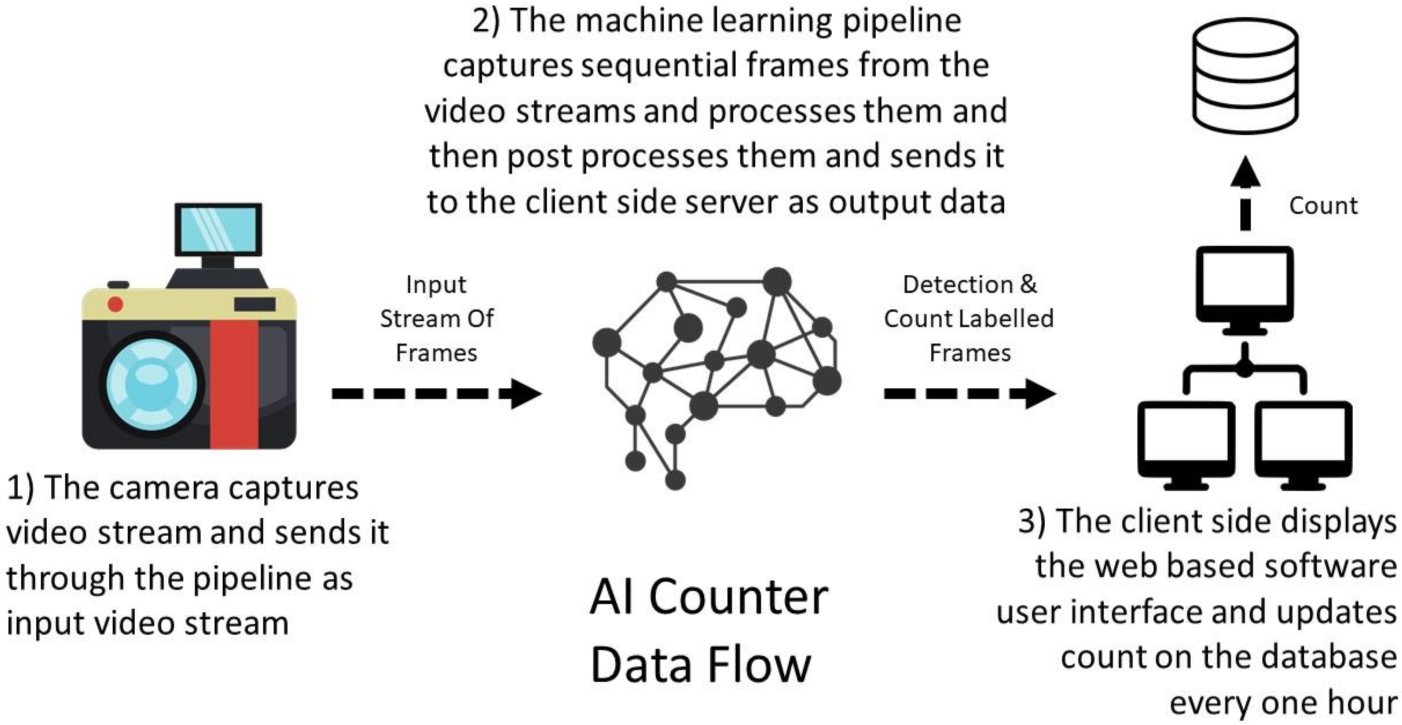

2) The machine learning pipeline captures sequential frames from the video streams and processes them and then post processes them and sends it to the client side server as output data
Count
Input Stream Of Frames
Detection & Count Labelled Frames
1) The camera captures video stream and sends it through the pipeline as input video stream
AI Counter Data Flow
3) The client side displays the web based software user interface and updates count on the database every one hour

Figure 1: AI Counter Data Flow and Structure Diagram

The implementation that we designed for this AI based conveyer belt classified counting was basically a machine learning pipeline based on image processing. From one end there is a camera which keeps sending frames every small fraction of a second, these frames are then inputted into the model which then does 4 main tasks:

1) localizes targeted object in the frame by drawing a rectangular bounding box around it

2) Classifies the localized objects

3) Tracks the object’s motion and direction through out the series of the frames in which the object is still detected in the image

4) raises a flag for classified class call once the object passes a certain threshold which then updates a dictionary that’s keeping track of the objects by classifying them and counting them

according to their classes. The output is then labelled in the frames and sent as output to the client side web based user interface which then updates the count on the database every one hour.

The **programming frameworks** used to build this project are:

- Python for base programming language
- Open-CV-python for video input output and cv2.dnn module for object detector model deployment [A2]
- Personally custom built dlib correlation tracker object class for object tracking [6]
- CUDA and CUDNN used for GPU acceleration [7]
- Flask Socket IO module for socket programming to connect backend processing to front end representation [8]
- YOLO darknet framework for training YOLO V3 model.[9]
- LabelImg for image labelling for YOLO V3 model training. [10]

Hardware used for prototype testing and deployment for **Proof of Concept** at industry compound:

- IP camera (Hikvision DS-2CD1023G0-I 2MP Basic IR Mini Bullet IP-Camera)
- C Panel in Industry Server for front end deployment
- CUDA compatible GPU (ZOTAC Gaming GeForce RTX 3060 Ti Twin Edge 8GB Graphics Card)
- High powered CPU (AMD Ryzen 5 5600X Processor)
- Compatible Motherboard (Asus Tuf Gaming X570 – Plus (Wi-fi))
- 16 GB DDR4 RAM (2 x 8 GB DDR4 RAM)

Now that we have discussed about the materialistic infrastructure of the project, lets talk more about it’s tasks and how our technology performs them:

[6] http://dlib.net/dlib/image_processing/correlation_tracker_abstract.h.html

[7] https://developer.nvidia.com/cuda-toolkit

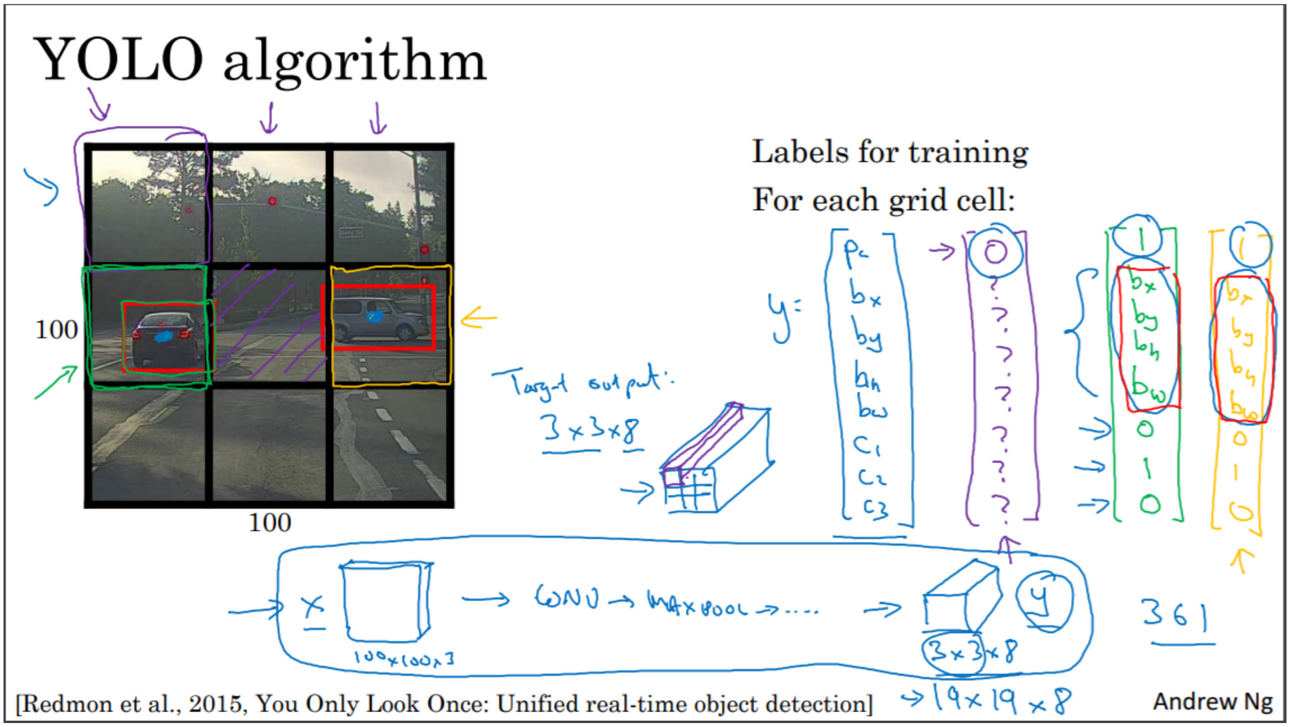
[8] https://github.com/ankur-arch/realtime-video-streaming-flask

[9] https://pjreddie.com/darknet/yolo/

[10] https://github.com/tzutalin/labelImg

5.2 Task 1 & Task 2

The first two tasks are done by the YOLO algorithm. The YOLO algorithm as mentioned in my literature review, consists of a series of 5+n output vectors in which n is the number of classes. Say for example we are detecting three categories of cars on the road. The Yolo algorithm would first take say, a 100x100x3 sized image and then after going through the hidden layer we wouldget a more concentrated output image that is 3x3x8 in size where 8 is the vector of output nodes (5 localization points and 3 class sparse vector) as shown in Figure 1.


YOLO algorithm

Labels for training
For each grid cell:

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

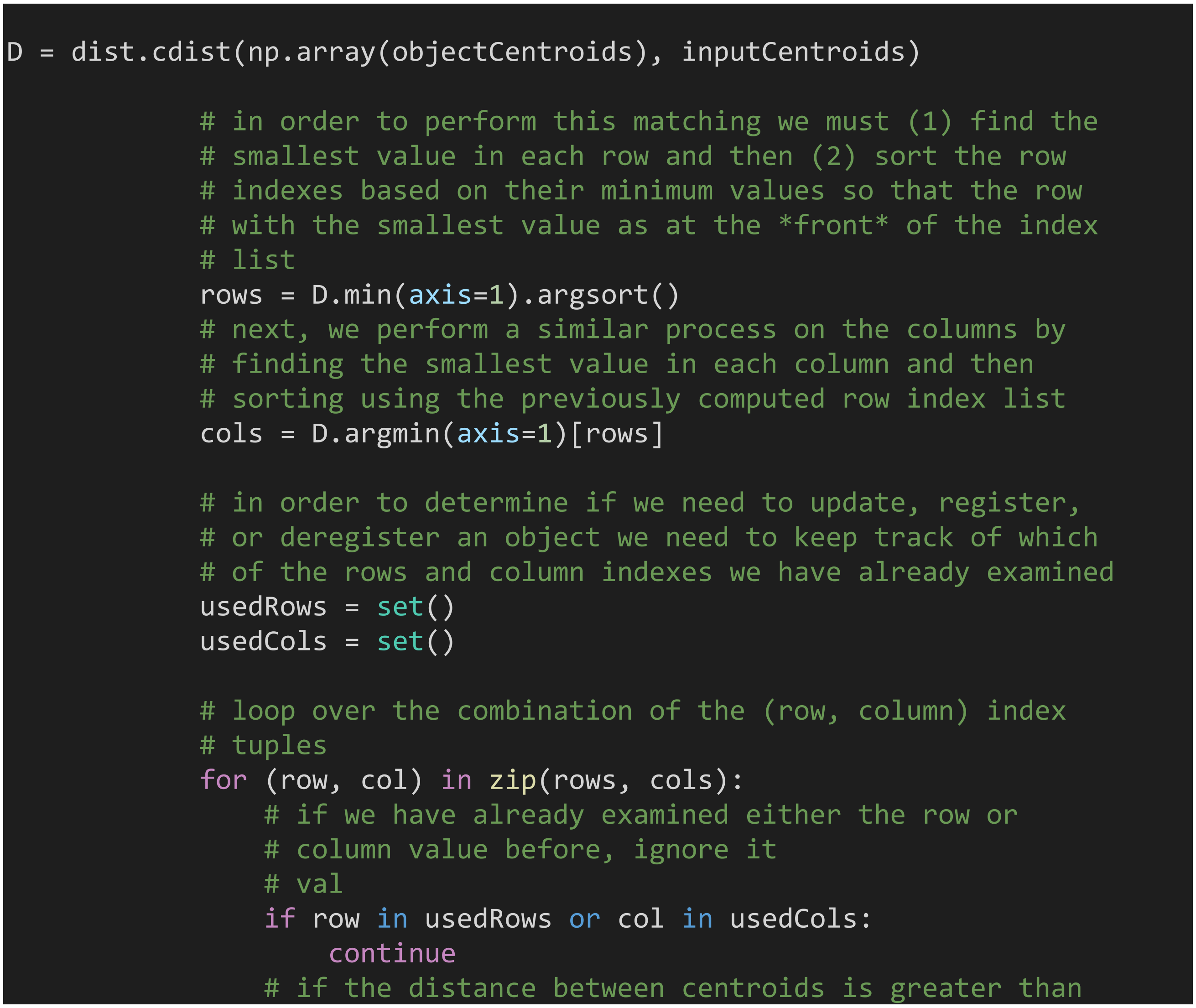
Andrew Ng

Figure 2: YOLO Output Node Architecture [Credit: DeepLearning.Ai]

As shown in the figure, the Pc is our first component which gives us an overall probability of whether there is an image in this region of the image or not. So in the figure, if we look at the 3x3 segmented picture we can estimate that only the left most and right most cells in the middle row will have a Pc with a value greater that 0.5 since cars are only on those two regions of the image. Bx and By provides us the coordinates of the middle point of the object as a ratio of the dimension of the image. Accordingly, Bh and Bw provides us with the size of the height and width of the object in them image as ratio to the height and width of the image itself. The red rectangles we see around the cars is what we call a graphical representation of localization (Task 1). As for the three categories, C1, C2 and C3, they are binary vectors which is used to classify the image (Task 2).[11]

5.3 Task 3 & Task 4

As for **Task 3 & 4**, that was done by our learning discriminative correlation filters. Even though the tracking mechanism seems very complicated mathematically, it was quite easy to apply as there was already a C++ implementation of it in dlib. The tracker helped us accomplish one main task and one secondary task. The main task was to map the detected objects according to their IDs from frame to frame with pairwise distances calculated between the Bx and By points of each detected boxes from the previous frame and the Bx and By points of each detected boxes in the present frame.



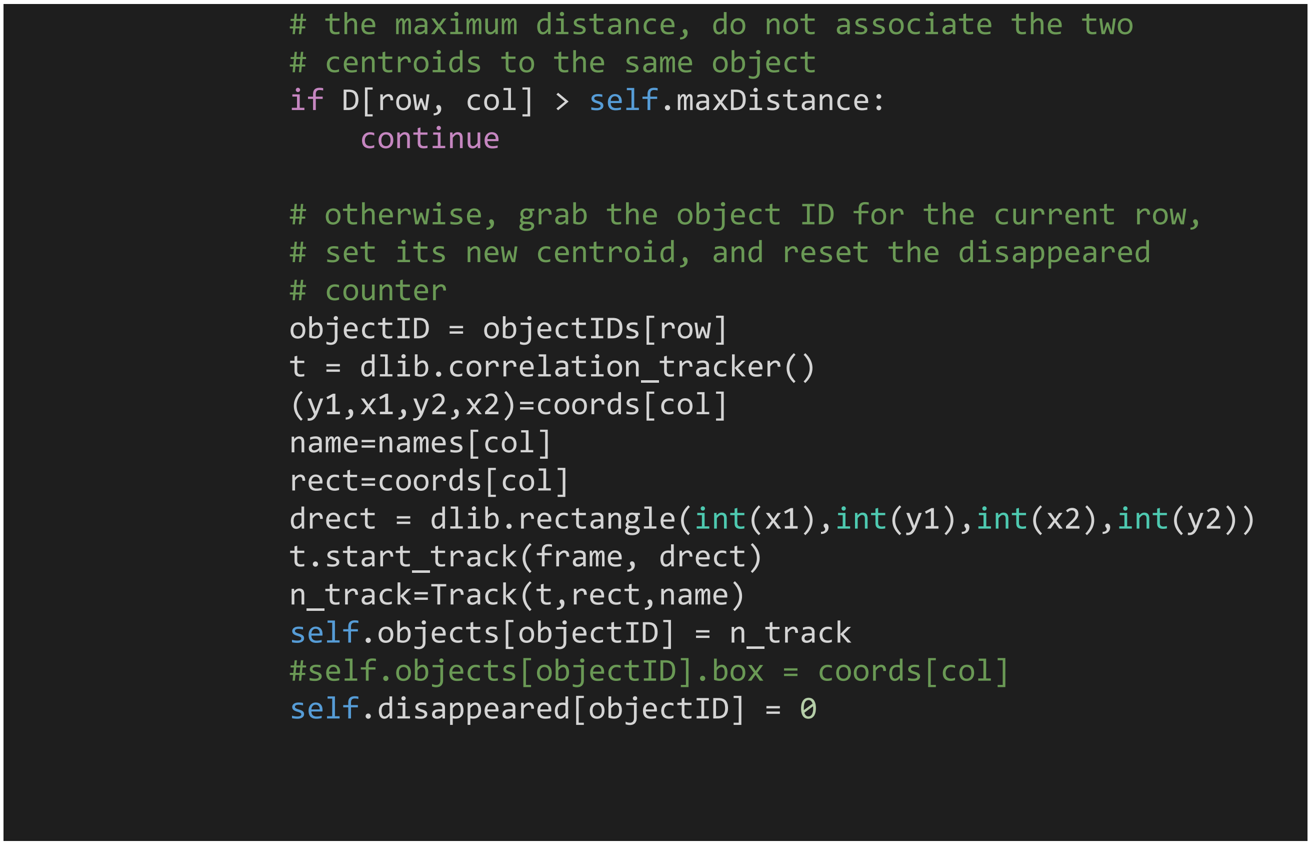
```
D = dist.cdist(np.array(objectCentroids), inputCentroids)

            # in order to perform this matching we must (1) find the
            # smallest value in each row and then (2) sort the row
            # indexes based on their minimum values so that the row
            # with the smallest value as at the *front* of the index
            # list
            rows = D.min(axis=1).argsort()
            # next, we perform a similar process on the columns by
            # finding the smallest value in each column and then
            # sorting using the previously computed row index list
            cols = D.argmin(axis=1)[rows]

            # in order to determine if we need to update, register,
            # or deregister an object we need to keep track of which
            # of the rows and column indexes we have already examined
            usedRows = set()
            usedCols = set()

            # loop over the combination of the (row, column) index
            # tuples
            for (row, col) in zip(rows, cols):
                # if we have already examined either the row or
                # column value before, ignore it
                # val
                if row in usedRows or col in usedCols:
                    continue
                # if the distance between centroids is greater than
```

[11] https://www.coursera.org/learn/convolutional-neural-networks/lecture/9EcTO/bounding-box-predictions

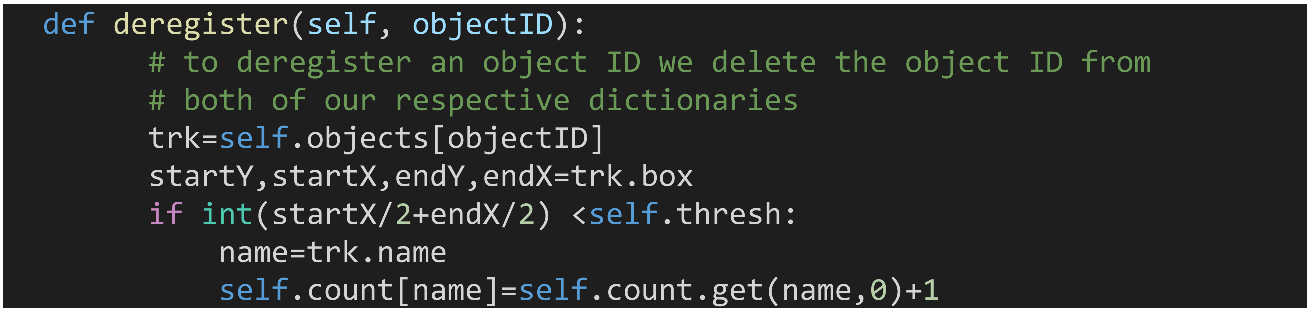


```
                # the maximum distance, do not associate the two
                # centroids to the same object
                if D[row, col] > self.maxDistance:
                    continue

                # otherwise, grab the object ID for the current row,
                # set its new centroid, and reset the disappeared
                # counter
                objectID = objectIDs[row]
                t = dlib.correlation_tracker()
                (y1,x1,y2,x2)=coords[col]
                name=names[col]
                rect=coords[col]
                drect = dlib.rectangle(int(x1),int(y1),int(x2),int(y2))
                t.start_track(frame, drect)
                n_track=Track(t,rect,name)
                self.objects[objectID] = n_track
                #self.objects[objectID].box = coords[col]
                self.disappeared[objectID] = 0
```

Code Basis of Centroid Distance Sorting

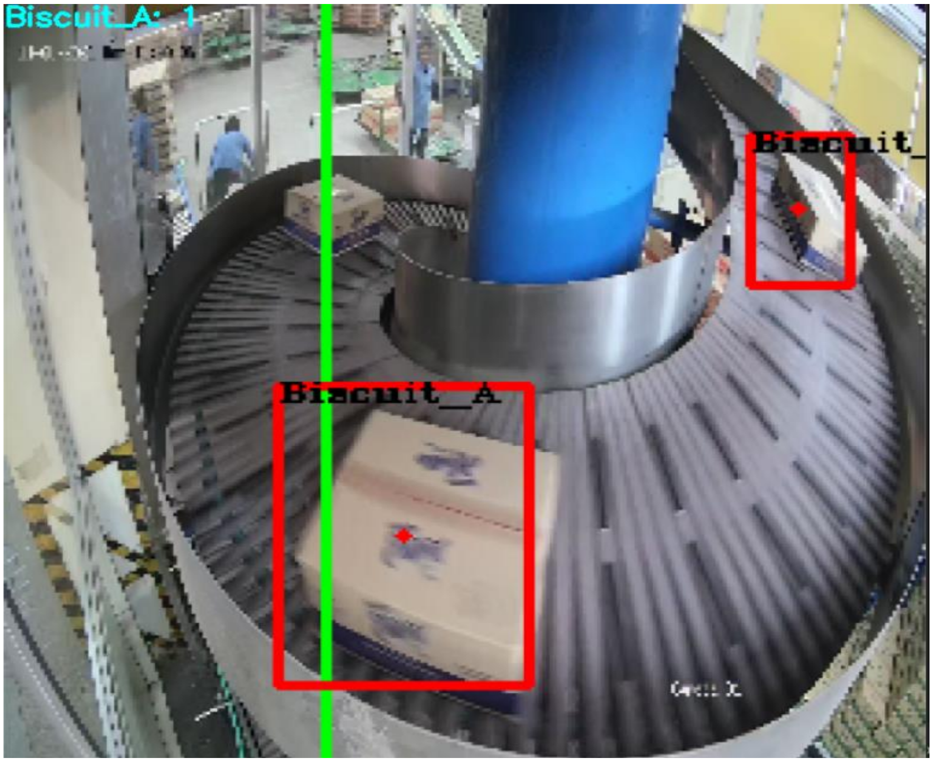
As we can see from the code, when we have an equal number of detected boxes in both now and previous frame, we sort out the pair wise distances between all combinations of different pairs of old and new centroids. This helps us create a pairwise coordinate based distance matrix (**D**). Then we loop over each of the combination of coordinates and the distances between them (between **old and new centroids**; **zip(rows,cols)**) and then see if they pass a certain distance threshold (**self.maxDistance**), in the case they do, the new coordinates are too far from the old coordinates in the pair to be the same objects. However, in the case the distance is lower, the ID from the old track object is assigned to a new track object with the new coordinates (**self.objects[objectID]=n_track=Track(t.rect,name)**). Like this, the IDs from previous frames are assigned to the newly detected objects and thus keeping a track of the flow of objects in the video. This way, once a particular object passes a threshold in the frame, we can easily count it.



```
def deregister(self, objectID):
      # to deregister an object ID we delete the object ID from
      # both of our respective dictionaries
      trk=self.objects[objectID]
      startY,startX,endY,endX=trk.box
      if int(startX/2+endX/2) <self.thresh:
          name=trk.name
          self.count[name]=self.count.get(name,0)+1
```

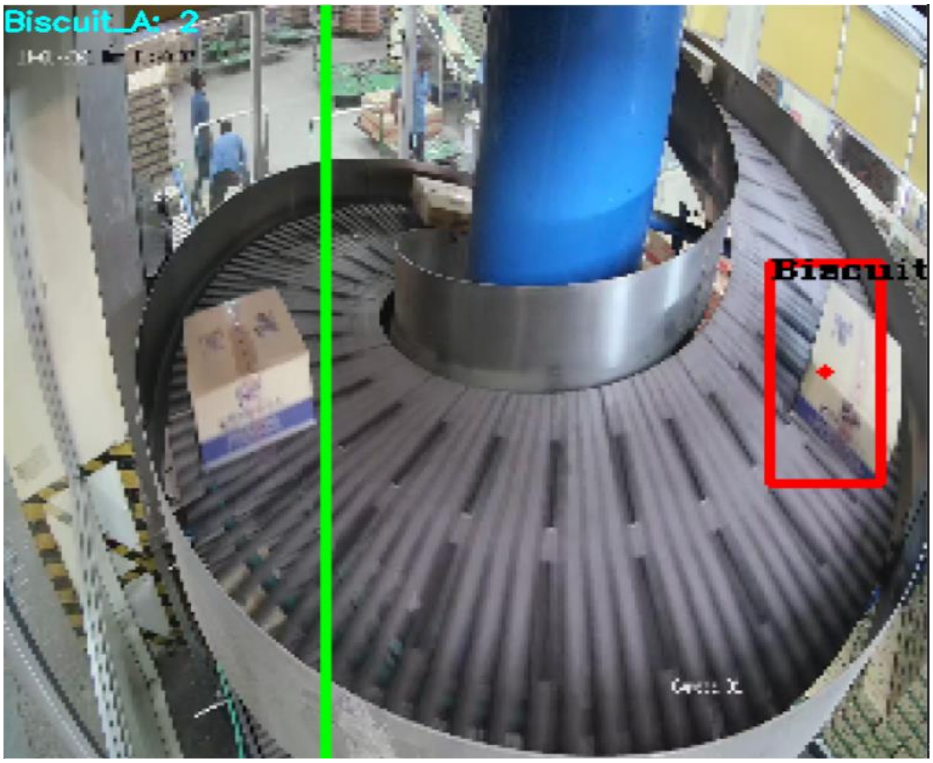
Code Basis for counting

As you can see, once the object passes through a certain threshold (**self.thresh**), we take its class name and call it in the dictionary (**self.count[name]**) and then increase its value by 1 (**self.count.get(name,0)+1**) thus counting the object. A two-picture demonstration of the code is given in the next page.



As we can see, before passing the green threshold, the object count of Buscuit 1 is still 1

Figure 3: Threshold based counting in video (Frame 1)



And as soon as the red dot in the box passes this **green line** (a graphical representation of **self.thresh**), the count is increased by 1

Figure 4: Threshold based counting in video (Frame 2)

The secondary task that our tracker fulfills is that it allows frames to be skipped from detection. It can predict the coordinates of the previous frames with its discriminative correlation filters, this way we can only detect frames using YOLO every say, 10 frames, or 5 frames, or 15 frames, depending on the speed of the objects required to keep the prediction of the localizations accurate. This way our total required computation time for the YOLO algorithm is divided by the number of frames skipped between every object detection. [6]

Chapter 6:

Data Analysis and Expected Investment Returns

Most of our data analysis and findings were based on our analysis of collected training data and their distributions along with tested accuracy in counting. We are presently at the final stages of our Proof of Concept testing. Till now we have trained the YOLO V3 model to classify 3 different ranges of categories, names of which shall only be written as Biscuit A, Biscuit B, Biscuit C. etc instead of their actual brand names due to privacy agreements with the company's clientele.

The model was trained on sets of 3 categories, 9 categories and 21 categories. Each of these three models provided us with significantly diverse accuracies so as to provide us with insight on why it is as such.

Table 1: Training data for 3 categories

Category Name:	Number of boxes annotated in frames
Biscuit A	1137
Biscuit B	1048
Biscuit C	963

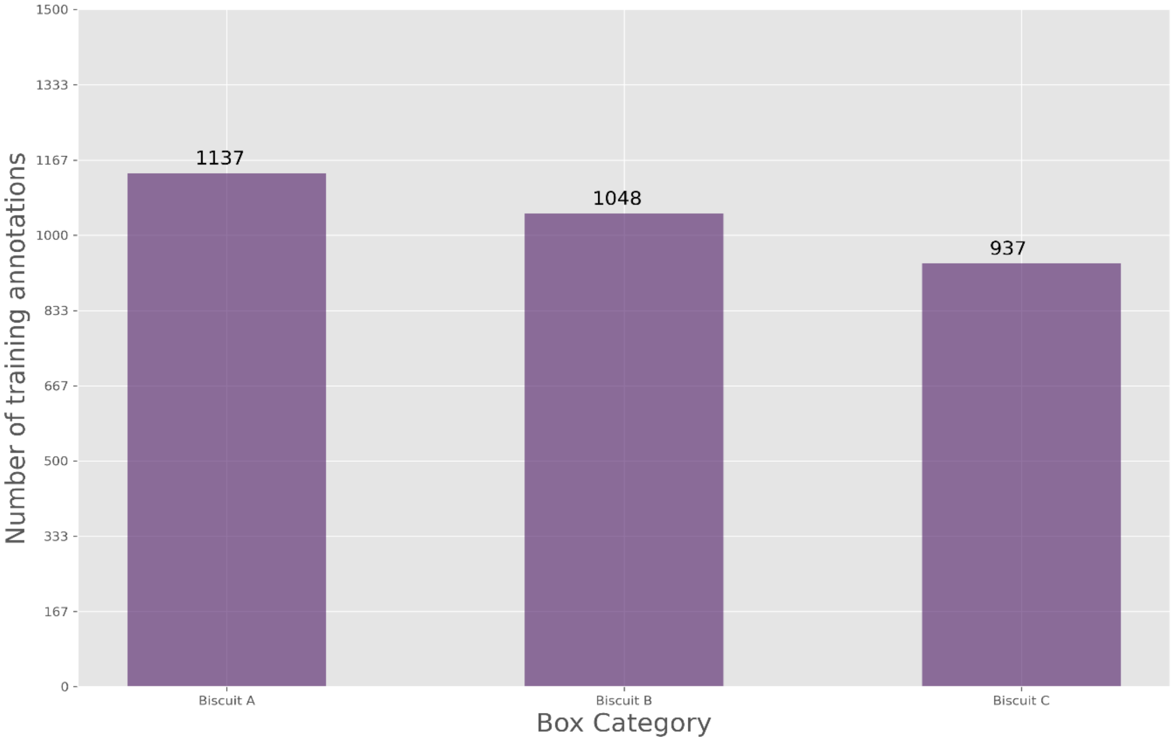


Figure 5: Bar Plot of Training Data for 3 box categories

The training data for the first three categories included annotations based on both disparate classes as well as multiple sequences of difficult box orientations (standing, one on top of another, slanting and upside down), Sequential data had to to be stored in similar volume per sequence for every class in order to get a balance distribution of training data per class.

6.1 Findings in training the YOLOV3 on the first 3 category dataset

The average accuracy achieved in the overall counting due to the proper classification and localization of the objects was about **99.1% percent with the model missing only 1-2 boxes in every 100 boxes in the case of very challenging box positioning in camera view and 100% accuracy in the case of spaced out and plainly positioned boxes in camera view.**

Table 2: Training data for 9 categories dataset:

Category Name:	Number of boxes annotated in frames
Biscuit A	670
Biscuit B	718
Biscuit C	652
Biscuit D	235
Biscuit E	311
Biscuit F	278
Biscuit G	223
Biscuit H	219
Biscuit I	256

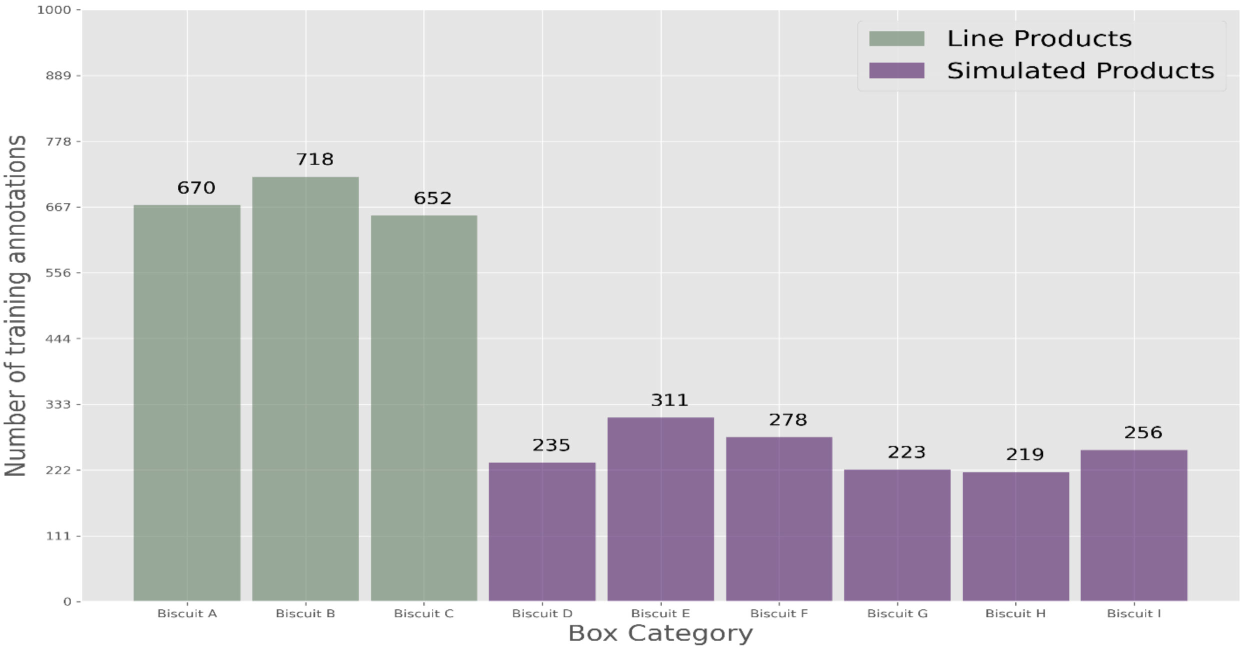


Figure 6: Bar Plot of Training Data for 9 box categories

6.2 Findings from creating the 9-category dataset and training YOLO on it

By the time we started working on the 9-category dataset, we found out that the first three categories we used were the only **line products** so the rest of the available products (which were not currently in production) had to be **simulated** through passage of old boxes of the other categories in the camera view of the conveyer belt. The problem with this is, the line products cannot be reduced or halted for the purpose of dataset creation as those products are following a production schedule. This is why each frame has an imbalance of annotated objects. For example, say ,we have 2 out-of-line products going in front of the camera's range of view; unfortunately, more often than not, there will also be a few line products coming along, this causes a distinct imbalance of annotated data between Line Products and Simulated products as shown in the figure above. Upon repetitive testing, we find the accuracy of this model to drop by an astounding **20 percent**, reaching an accuracy of **75%** when it comes to real-time classified counting.

Table 3: Training data for 21 categories dataset

Category names	Number of boxes annotated in frames
Biscuit A	1764
Biscuit B	799
Biscuit C	91
Biscuit D	141
Biscuit E	125
Biscuit F	264
Biscuit G	436
Biscuit H	307
Biscuit I	127
Biscuit J	254
Biscuit K	245
Biscuit L	228
Biscuit M	143
Biscuit N	199
Biscuit O	186
Biscuit P	154
Biscuit Q	171
Biscuit R	213
Biscuit S	267
Biscuit T	351
Biscuit U	178

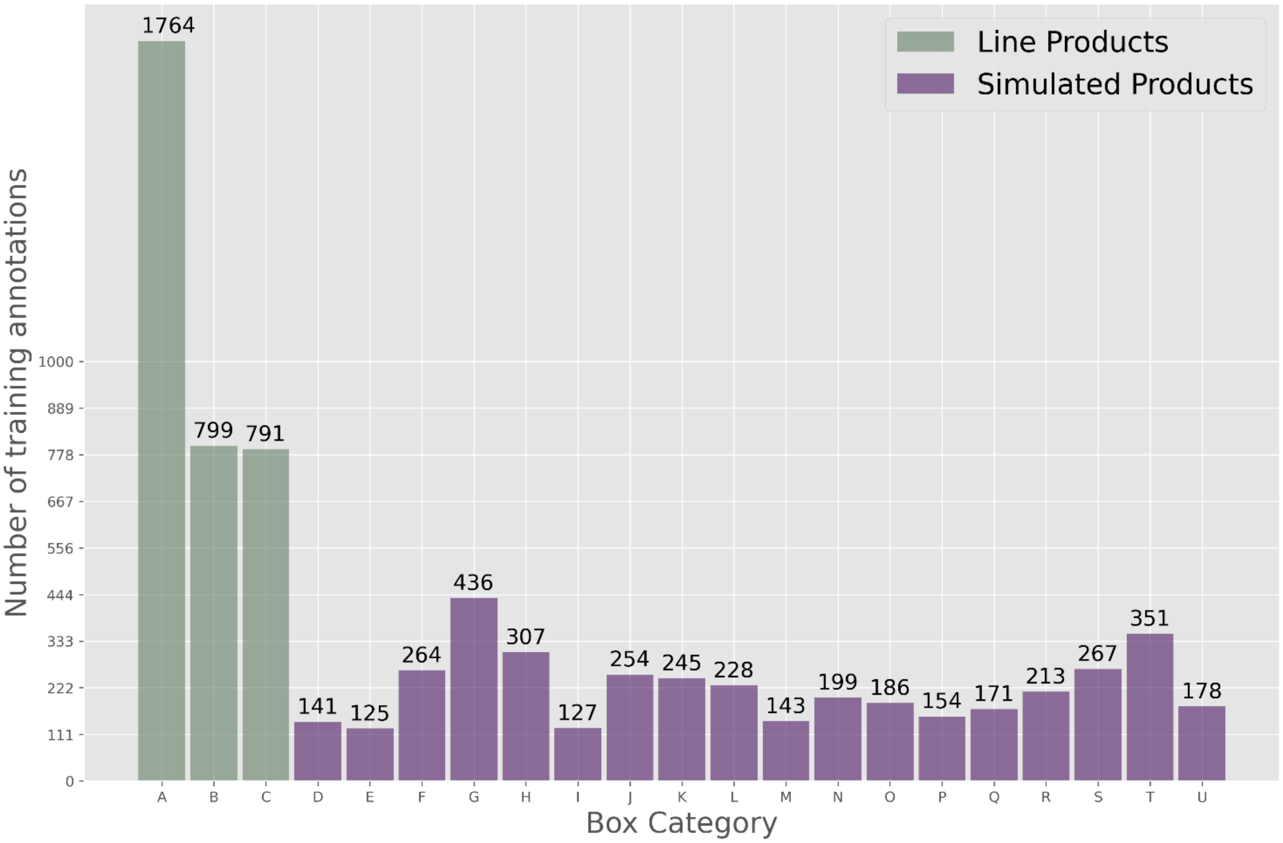


Figure 7: Bar Plot of Training Data for 21 box categories

6.3 Findings from creating the 21-category dataset and training YOLO on it

In training the 21-category model we find further difficulties with data preprocessing as we find that the class imbalance in category count has gone up even more. The main problem falls unto the human error of the assistants on the conveyer belts who helped us simulate passage of boxes not scheduled for immediate production in equalizing distribution of class frequency. Upon repetitive testing we find that that the accuracy in classified counting has dropped to only a mere **29%**.

6.4 Results and Insights gained from training data and test accuracies

As we can see from the three phases of training, there is a **clear relation** between the **degree of imbalance in class distribution of training data for algorithm** and **its accuracy in application**, which in this case, is classified **counting/tallying**. So in order to find a proper scientifically representational metric to evaluate this relation, we shall define our imbalance of data in mathematical terms.

We know that:

Our imbalance metric needs to be great when there is lots of imbalance in each of the classes. In order to achieve this we shall be implementing our own Imbalance Score for all three training phases.

Let the total number of classes be n

Let, the standard number of training data per class for an algorithm = X

Let, the actual number of training data per class = x'

Then we can find the absolute class imbalance in ratio to the actual standard product for each class as

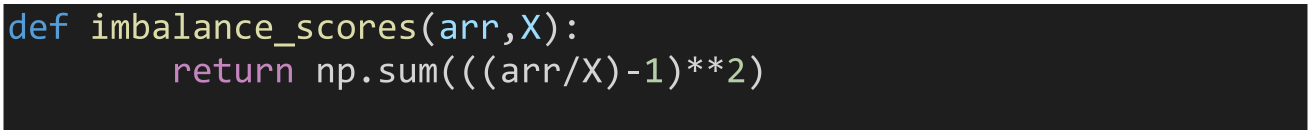
$(\frac{{x'}_k}{X} - 1)^2$

Therefore our imbalance score for each training phase can be denoted as:

$$= \sum_{k=1}^{n} (\frac{{x'}_k}{X} - 1)^2$$

Note: we used summation method because we noticed that in real life deployment applications, imbalance in each category creates its own localization issues when the model is integrated with the tracker.

In Python coding syntax the formula would look like:



```
def imbalance_scores(arr,X):
        return np.sum(((arr/X)-1)**2)
```

In the case of YOLO Algorithm, the standard amount of images per class along with special sequence cushion count is 800. Therefore:

Table 4: Imbalance scores for three training phases

Training Phases	Imbalance Scores	Counting Accuracy
3 Categories	0.31	0.991
9 Categories	2.879	0.75
21 Categories	11.044	0.29

As we can see there is a clear relationship between the Counting Accuracy and the Imbalance Scores and if we were to fit this data into a linear regression optimized by Stochastic Gradient Descent, we get:

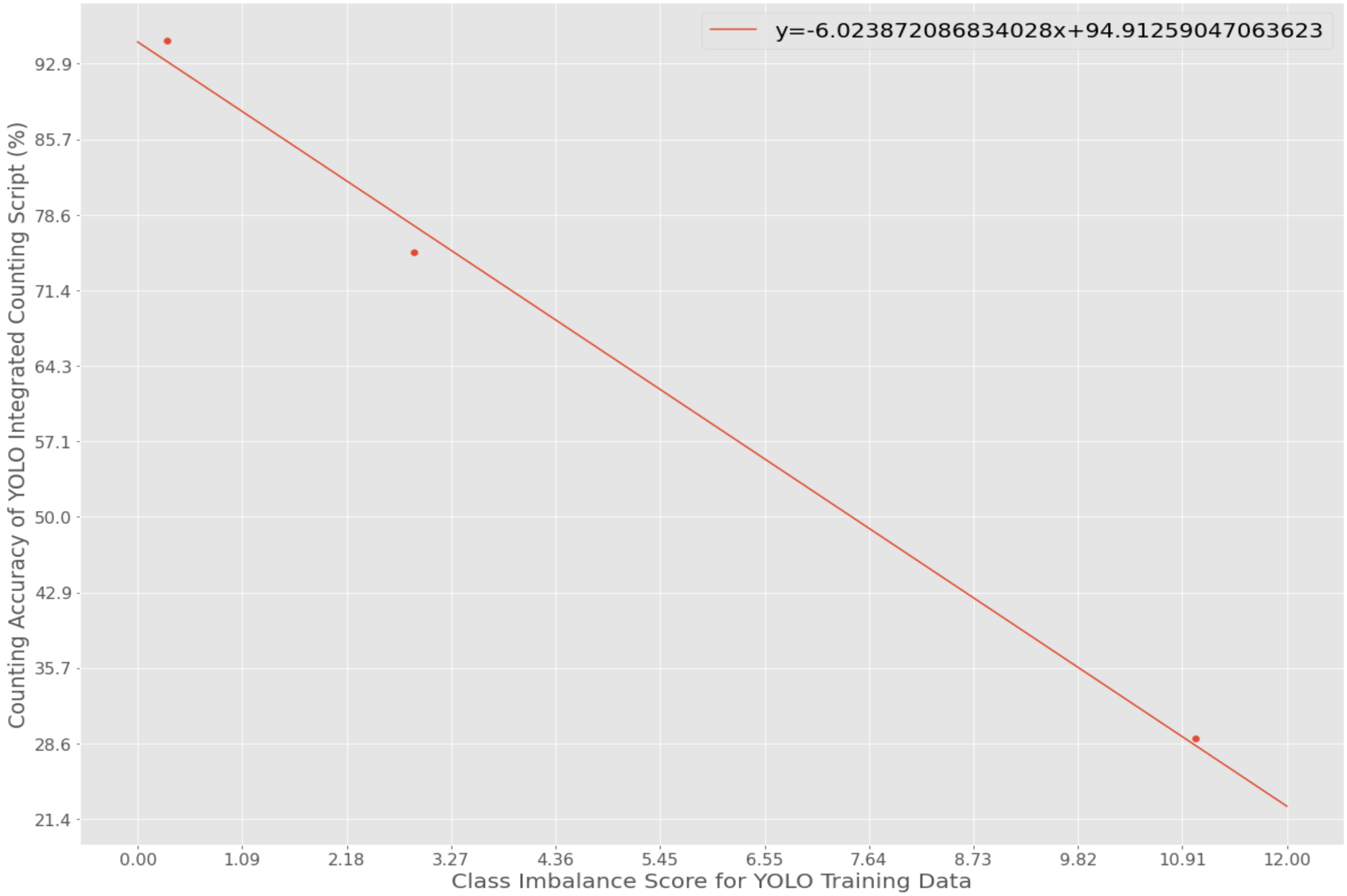


Figure 8: Line and Scatter Plot of Regression line in Accuracy Vs Imbalance Metrics

As we can see **the regression line** is **downward sloping** with a **slope coefficient of -6.02**. This means that with **every one unit increase of Class Imbalance Score**, our **Counting Accuracy maybe subjected to go down by about 6 units**. Thus, along with the graphical representation and the evaluation of the slope value of the regression model. It is hence proved that there is **a strong negative correlation** between Class Imbalance in Training and Counting Accuracy of the model, specifically, **-0.997639**. [A3 & A4]

6.5 Investment Return Period: (Breakeven Analysis)

According to our hardware requirements, we already know the investment cost of our Client Industry in implementing the Computer Vision based Counter:

Table 5: Price Quotation of Client Industry's hardware expenditure and monthly variable expenditure on deployed model

Details	**Price (Tk)**
2 x IP camera (Hikvision DS-2CD1023G0-I 2MP Basic IR Mini Bullet IP-Camera)	2 x 7000
C Panel in Industry Server for front end deployment	5000
CUDA compatible GPU (ZOTAC Gaming GeForce RTX 3060 Ti Twin Edge 8GB Graphics Card)	47000
High powered CPU (AMD Ryzen 5 5600X Processor)	31000
Compatible Motherboard (Asus Tuf Gaming X570 – Plus (Wi-fi))	27000
16 GB DDR4 RAM (2 x 8 GB DDR4 RAM)	15000
Transportation and Miscellaneous Expenses	50000
Total Fixed Cost	**189000**
Variable Expenses: Extra Electricity + Internet cost for Deployed Model	20,000/month

As we can see from the above Table:

- Due to the high computational resource requirements of YOLO, only two camera counters could be powered by the available computational power provided by our specialized hardware.
- AI Counter Deployment's Fixed Expenditure is Tk 189,000 /=
- Our Variable Expenses is Tk 20,000 /=
- Due to the automation of the counting task in the industry, two tallying employees, one from each 12 hours shift, shall no longer be needed per conveyer belt the counter camera is deployed on. Therefore, variable expenses saved per month would amount to Tk 10,000 x 2 x 2 = Tk 40,000/=.
- Therefore, the company would reach its breakeven point from this investment within:

$$\frac{189{,}000}{40{,}000-20{,}000} = 9.45\ month = 9\ months\ and\ 14\ days$$

As calculated, the company could **earn back its expenses within a year** and **earn about: (12-9.5) x 20,000) = 50,000 Tk worth of profit from the expenses saved** due to implementation of our technology.

Chapter – 7
Conclusion and Recommendation

7.1 Conclusion

The YOLO model is computationally quite expensive, but even while taking into account the high electricity and dedicated internet line bills, the model would perform quite well on industrial conveyer belts that have less than 6 categories of items passing through them at a time. Even better accuracy could have been reached in classified counting if a conveyer belt could have been isolated solely for the building of the training dataset for the YOLO V3. But in reality Industries are usually in full throttle to meet production quotas all the time and given how dynamic and fast paced Industry 4.0 is in reality, it is near to impossible to isolate a conveyer belt solely for training and building datasets with proper class balance.

The pipeline scripts are usually smoothly built within a month or two of R&D but the biggest difficulties faced in implementation of AI in Food & Beverage Industries of Bangladesh is from the huge number of data security and access protocols that need to be followed before access of required data is made available. Another big concern for AI implementation in Bangladesh is that Food and Beverage companies are still very hesitant about cloud based implementation and the industry is still very resistant towards the changes brought by cloud technologies. This requires implementation of locally made socket programs for data passage, which are not usually as optimized as those provided by commercial cloud architectures like Amazon Web Services or Microsoft Azure. Overall, there is still a huge space for the improvement and optimization of such tracking based Computer Vision Counters.

7.2 Recommendation

As we can see above, there are quite a few points of improvements required for the YOLO based Counter. Some recommendations would be:

- The **computational bottleneck** of this model is within the **processing needs of the YOLO model**, which **goes through 105 layers of convolutions before reaching the end of one forward pass**. So, instead of opting for **a single stage localization and classification model** like YOLO, **much less computationally expensive pairs of localization models** (e.g. Single Shot Detectors, a 22 layered object detector that has a 22 microsecond per frame processing speed on Tensorflow Object Detection API [12]) **and classification models** (e.g. VGG16, a 16 layered classification model that reached 93% accuracy in ImageNet dataset, a dataset of 1000 classes and 10 million image data points) are available with proper training and implementation documentation. Although the scripting might be a little complex for an ensemble image acquisition-based model (networks than extract images from detection boxes before classifying them), we believe that this kind of model would reduce the computational requirements of the technology's single operation cycle to a fifth of its current computational requirements. This would mean we could add five more cameras to 5 more conveyer belts and thus save five times the inventory tallying wages with only our existing computational resources. Our team is currently doing R&D on scripting this type of a model.

- Also, by using Image Acquisition model we can modify where we want the detected objects to be classified according to convenience of view and camera angle, thus giving us less training data requirements for the object detector model, lots of balanced data for classification model training as well as much greater control over the operations in the software
- In the case competent labor isn't available for developing an image acquisition-based system, the YOLO model could still be used for simplicity by resizing the input frame size to 250 x 250 dimensions from the existing 416 x 416 norms. This way the computational requirements could be brought down enough to add a third camera and thus save 20000 Tk extra at the small expense of retraining the YOLO model with 250 x 250 dimensioned input nodes and slightly lower accuracy.
- Another way to influence our computational bottleneck is by educating investors on the benefits of Cloud Technology in order to influence them to invest in Commercial Cloud Based Deployment, that way, more efficient pipelines and cheaper computational resources could be used to provide greater returns to firms at the same existing cost of implementation.

[12] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Chapter – 8
References

[1] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, 20-25 June 2005, Vol. 1, 886-893. https://doi.org/10.1109/CVPR.2005.177

[2] Redmon, J. and Farhadi, A. (2018) YOLO v3: An Incremental Improvement. arxiv:1804.02767v1 [cs.CV], Unpublished.

[3] YOLOv4: Optimal Speed and Accuracy of Object Detection; Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao; https://arxiv.org/abs/2004.10934

[4] Yolov5-based channel pruning is used for real-time detection of construction workers' safety helmets and anti-slip shoes in informationalized construction sites; September 2021, Journal of Physics Conference Series 2031(1):012027; DOI:10.1088/1742-6596/2031/1/012027

[5] [**Danelljan et al**., BMVC **2014**, PAMI 2017]: https://conferences.mpi-inf.mpg.de/dagm/2018/template/GCPR_tracking_tutorial_181009_pdf_ver.pdfPowerPoint-presentation (mpg.de)

[6] dlib correlation tracker documentation: http://dlib.net/dlib/image_processing/correlation_tracker_abstract.h.html

[7] CUDA Toolkit: https://developer.nvidia.com/cuda-toolkit

[8] Flask Socket IO:

- https://github.com/ankur-arch/realtime-video-streaming-flask
- https://flask-socketio.readthedocs.io/en/latest/

[9] YOLO darknet framework: https://pjreddie.com/darknet/yolo/

[10] LabelImg: https://github.com/tzutalin/labelImg

[11] https://www.coursera.org/learn/convolutional-neural-networks/lecture/9EcTO/bounding-box-predictions

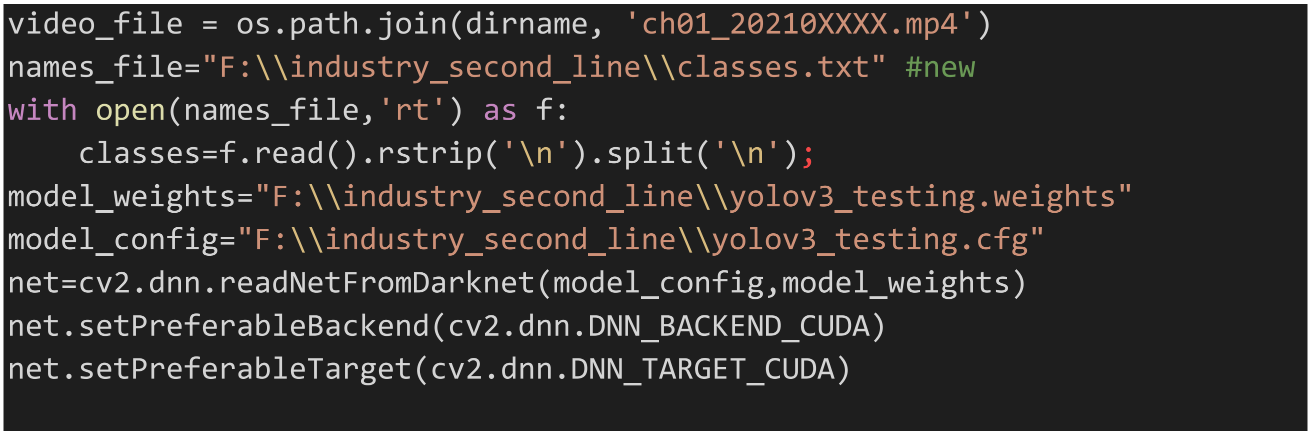
[12] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo

Chapter – 9
Appendix

[A1] Drive Link to Extended Testing Output Video (3 categories)

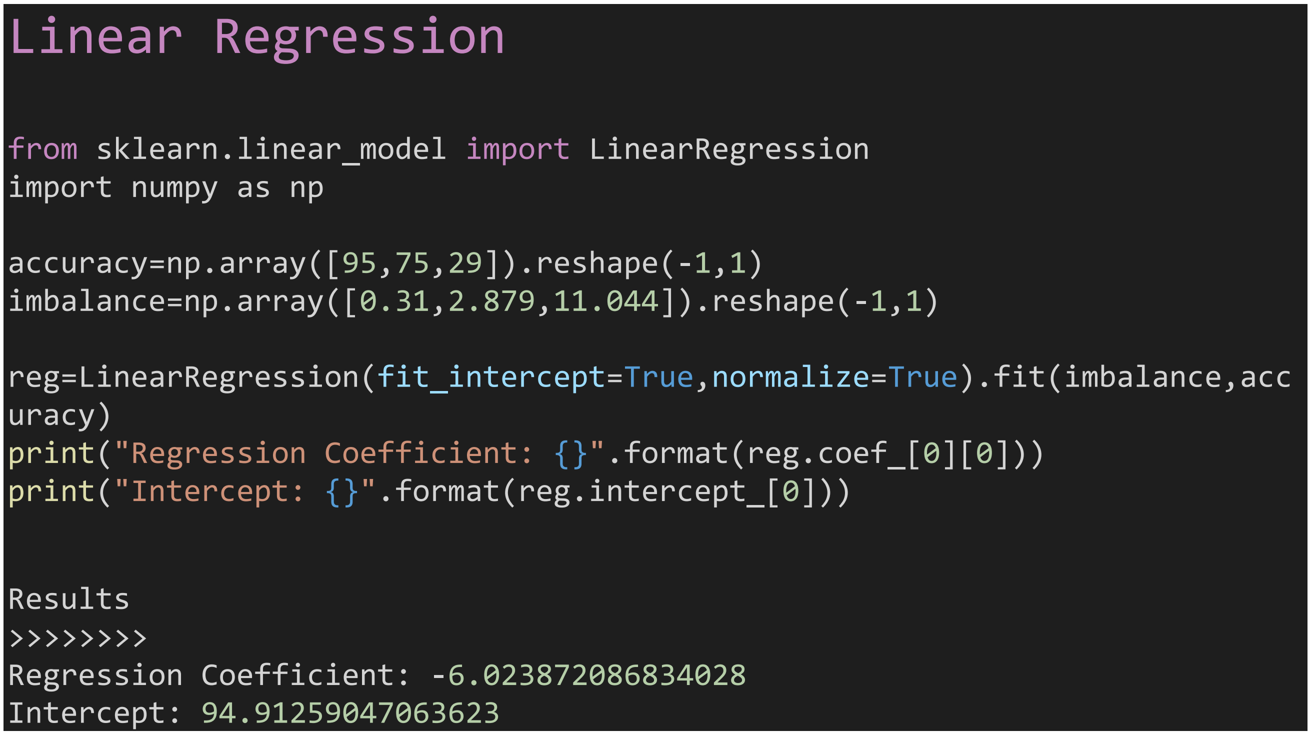
https://drive.google.com/file/d/1_2nCuA3ZFoR6glh_qn24BdQTrLRknLvM/view?usp=sharing

[A2] Deployment Codes for Trained model with Open CV



```
video_file = os.path.join(dirname, 'ch01_20210XXXX.mp4')
names_file="F:\\industry_second_line\\classes.txt" #new
with open(names_file,'rt') as f:
    classes=f.read().rstrip('\n').split('\n');
model_weights="F:\\industry_second_line\\yolov3_testing.weights"
model_config="F:\\industry_second_line\\yolov3_testing.cfg"
net=cv2.dnn.readNetFromDarknet(model_config,model_weights)
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)
```

[A3] Code reference for Linear Regression and Correlation Coefficient Calculation on Counting Accuracy vs Class Imbalance in Training Data



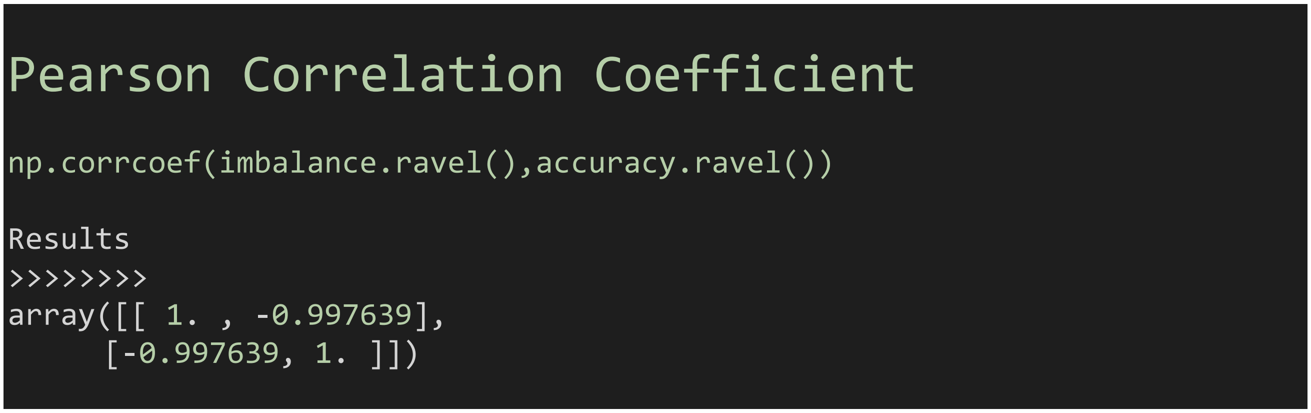
```
Linear Regression


from sklearn.linear_model import LinearRegression
import numpy as np

accuracy=np.array([95,75,29]).reshape(-1,1)
imbalance=np.array([0.31,2.879,11.044]).reshape(-1,1)

reg=LinearRegression(fit_intercept=True,normalize=True).fit(imbalance,acc
uracy)
print("Regression Coefficient: {}".format(reg.coef_[0][0]))
print("Intercept: {}".format(reg.intercept_[0]))


Results
>>>>>>>>
Regression Coefficient: -6.023872086834028
Intercept: 94.91259047063623
```

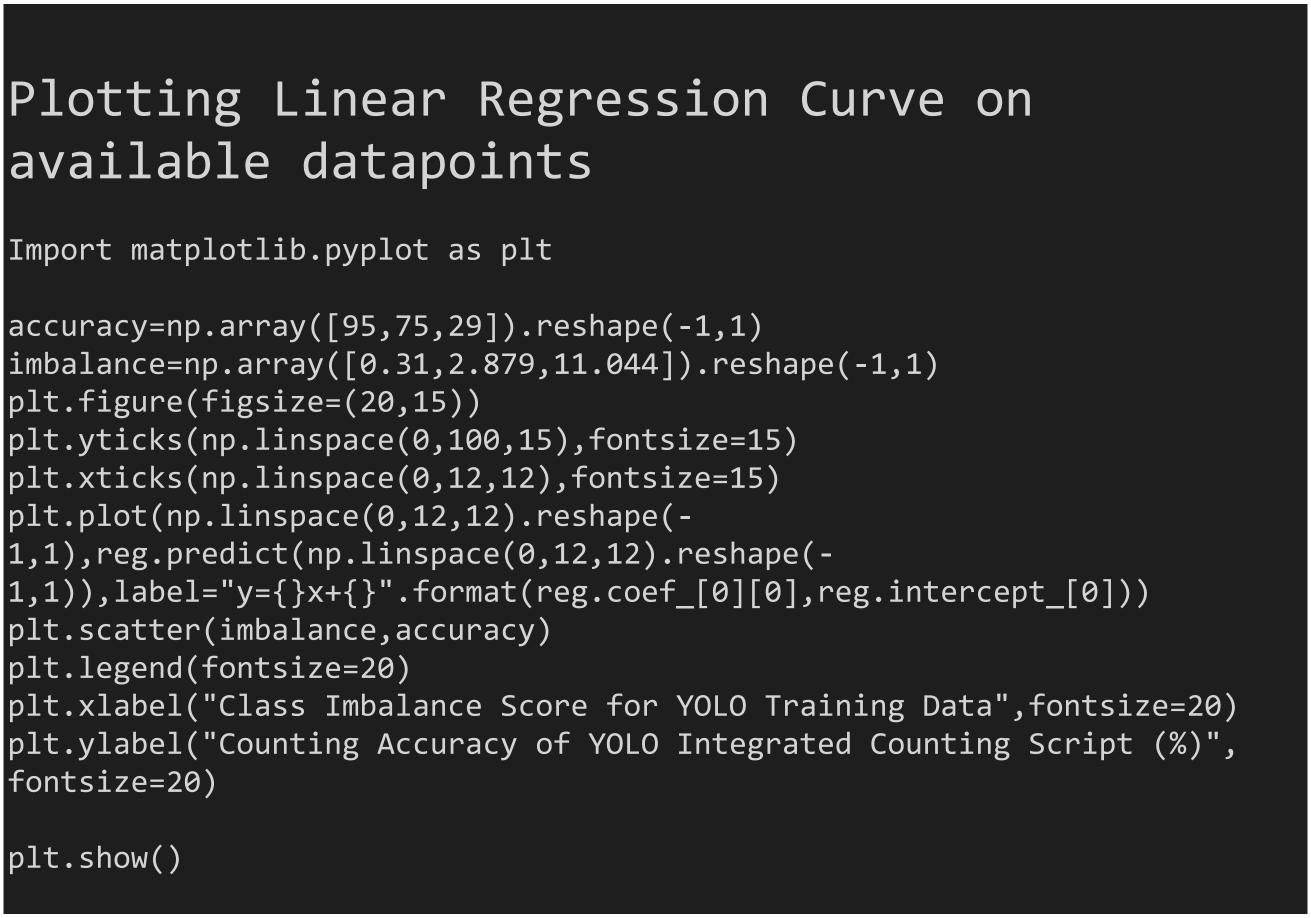


Pearson Correlation Coefficient

```
np.corrcoef(imbalance.ravel(),accuracy.ravel())

Results
>>>>>>>>
array([[ 1. , -0.997639],
      [-0.997639, 1. ]])
```

[A4] Code implementation for plotting regression curve



Plotting Linear Regression Curve on available datapoints

```
Import matplotlib.pyplot as plt

accuracy=np.array([95,75,29]).reshape(-1,1)
imbalance=np.array([0.31,2.879,11.044]).reshape(-1,1)
plt.figure(figsize=(20,15))
plt.yticks(np.linspace(0,100,15),fontsize=15)
plt.xticks(np.linspace(0,12,12),fontsize=15)
plt.plot(np.linspace(0,12,12).reshape(-
1,1),reg.predict(np.linspace(0,12,12).reshape(-
1,1)),label="y={}x+{}".format(reg.coef_[0][0],reg.intercept_[0]))
plt.scatter(imbalance,accuracy)
plt.legend(fontsize=20)
plt.xlabel("Class Imbalance Score for YOLO Training Data",fontsize=20)
plt.ylabel("Counting Accuracy of YOLO Integrated Counting Script (%)",
fontsize=20)

plt.show()
```

[A5] Internship Acceptance Letter

Dr. Md. Akram Hossain
Professor & Chairman
Department of Management Information Systems
Faculty of Business Studies
University of Dhaka
Nilkhet Road, Dhaka-1000.

Subject: Acceptance of Internship.

Respected Sir,

We are pleased to inform you that the Management of **Essential Infotech** has decided to accept your internship offer for BBA student of **Mohammad Raghib Noor**, Bearing ID- **29-12-128.** He started this internship under the Artificial Intelligence department on April 25th, 2021.

His training will be from 10.00 am to 6.00 pm, 05 days a week, Regularity will be expected from this intern during his 12 weeks tenure at the organization.

We wish him a successful internship at **Essential Infotech**. For any queries please feel free to contact us.

Best Regards

MD: Navid Riad
Marketing Head
HR & Marketing Department
Cell- +8801773-431798

[A6] Internship Completion Letter

Essential
Infotech

Dr. Md. Akram Hossain
Professor & Chairman
Department of Management Information Systems
Faculty of Business Studies
University of Dhaka
Nilkhet Road, Dhaka-1000.

Subject: Completion of Internship.

Respected Sir,

We are pleased to inform you that the Management of **Essential InfoTech** has deemed **Mohammad Raghib Noor**, bearing ID- **29-12-128,** competent in passing his internship evaluations**.** He started this internship under the Artificial Intelligence department on April 25th; 2021.and has completed his Internship on August 5th, 2021 by dutifully fulfilling his duties as a junior AI developer and as lead AI developer for a clientele proof of concept.

We wish him tremendous success in his future endeavours.

Best Regards,
Mostafizur Rahman Khan

Operational Head
Management and Finance
Cell- +8801739303737