SACK COUNTING SYSTEM LEVERAGING VIDEO ANALYTICS

Aastha Vashistha[1], Anant Singhal[2], Ritin Behl[3]

[1]*Student, Department of Information Technology, ABES Engineering College, Ghaziabad,Uttar Pradesh, India*
[2]*Student, Department of Information Technology, ABES Engineering College, Ghaziabad, Uttar Pradesh, India*
[3]*Professor, Department of Information Technology, ABES Engineering College, Ghaziabad, Uttar Pradesh, India*

---***---

Abstract- *This paper is an attempt at developing an object detection and counting system using modern computer vision technology. The paper proposes a sack detection and counting system. This project aims at reducing the human effort involved during tracking and counting of sacks for logistics management. A better technique to achieve this goal is using image processing algorithms and methods on warehouse camera video feed. This paper presents a sack detection classifier based on object detection algorithm and frameworks for image processing including object detection.*

Manual counting of sacks has been carried out but it takes a lot of time and requires more labor. TensorFlow Object Detection API[4], an open source framework for object detection related tasks, is used for training and testing a Faster RCNN[5] (Reinforcement Convolutional Neural Network) model. The model was tested as fine-tuning with a dataset consisting of images extracted from video footage of sack loading in truck. This system includes preprocessing of images, extraction of features, tracking of objects and counting using machine-learning algorithms. This paper presents computer vision and machine learning techniques for on sack detection, tracking and counting.

Keywords: Machine learning, object detection, TensorFlowObject Detection API, Faster RCNN model, image processing

1. INTRODUCTION:

Object detection is a technique which involves locating the presence of objects with a bounding box and types or classes of the located objects in an image. Any image with one or more objects, consisting of the desired object and other background as noise can be an input. The output would be a window with detected objects in the annotated box and confidence score of each detected object above it.

Object detection combines the two tasks of image classification and object localization, and localizes and classifies one or more objects in an image. When it comes to image classification, it is all about predicting the class of one object in an image. Also, object localization is defined as identifying the location of one or more objects in an image and drawing a bounding box or annotating the desired object in an image around them. To achieve better object detection, classification and recognition, different machine learning algorithms and feature extraction algorithms are used. The achievement of accuracy can be different for different algorithms. Hence, we need to select the (best) algorithm with the highest classification and prediction accuracy. Also, while selecting the algorithm it is important to decide between speed and accuracy factors. For sack detection and counting this project implements a portion of computer vision and object detection with machine learning model using tensorflow api. In recent years, computer vision has been applied in the transportation and logistics sector for better automation results.

1.1 NEED OF IMAGE PROCESSING IN LOGISTICS:

For detection of sacks in the warehouse should have to be taken by using a camera which takes live feed. Images are taken as input and to detect sack image preprocessing, removal of noise, image enhancement feature extraction, feature selection and classification has to be done by applying machine learning algorithms. Once the image is in good condition and readable to the machine, the features extraction processes could be used to acquire useful content from the image. Image processing is used to extract, recognize, and classify features. Techniques that may offer many options in results and accuracy are Convolutional Neural Network and Object detection algorithm. Often it happens, human observation is required to keep track of objects unloaded from the truck to the warehouse or loaded from the warehouse into the trucks. There is a possibility of errors. This project is developed to solve this problem in the area of logistics and transportation industry.

2. LITERATURE SURVEY:

Object detection generally refers to computer vision techniques that involve classification and object location in an image. The most famous and relevant object detection algorithms are heavily based on use of CNN or convolutional neural networks [1]. There are some famous and relevant system types today named as Faster RCNN(Reinforcement Convolutional Neural Network) algorithm, SSD(Single Shot Detector) and YOLO(You Only Look Once)[1]. Original RCNN method worked by running a neural net classifier on samples cropped from images using externally computed box proposals(i.e samples cropped with externally computed box proposals; feature extraction done on all the cropped samples). The given approach was computationally expensive in terms of speed and accuracy due to many things, Fast R-CNN reduced the computation by performing the feature extraction only once on the whole image and cropping on the lower layer that is, feature extraction only once on the whole image and then cropping of samples with box proposals computed on external basis. Faster RCNN[5] goes a step further and uses the extracted features to create class-agnostic box proposals, that is, feature extraction only once on the whole image; no externally computed box proposals. R-FCN is like Faster R-CNN but the feature cropping is done in a different layer for increased efficiency [1]. YOLO (You Only Look Once) [2] works on a different principle than the above mentioned models. It works by running a single convolutional network on the whole input image only once to predict bounding boxes with confidence scores or annotate the input image with confidence score and prediction box for each class simultaneously. The advantage of the simplicity of the approach is that the YOLO model is fast as compared to Faster RCNN and SSD and it learns a general representation of the objects. Due to this localization error rate increases and also, YOLO performed badly on images with new aspect ratios but false positive rate is also reduced. Single Shot Detector (SSD)[3] is different from the RCNN based approaches as it does not require a second stage of classification operation on every proposal [1]. This makes it better and faster for real-time custom object detection applications. However, due to this accuracy gets reduced on validation image[1].

SSD(Single Shot Detector) adheres to a model where the middle architecture of the model is SSD. When it comes to the tradeoff between speed and accuracy, many custom object detection applications require real-time speed. Some methods such as YOLO or SSD perform faster than others, but this comes with a fall in accuracy of predictions on validation data. On the other hand models such as Faster RCNN achieve high accuracy but they are more expensive in terms of memory to run. The cost of model speed depends on the application. With larger images with high resolution, SSD performs comparably to more computationally expensive models in terms of accuracy such as Faster RCNN, even as on smaller images with low resolution and size, its performance is lower as compared to other models [1].

3. METHODOLOGY:

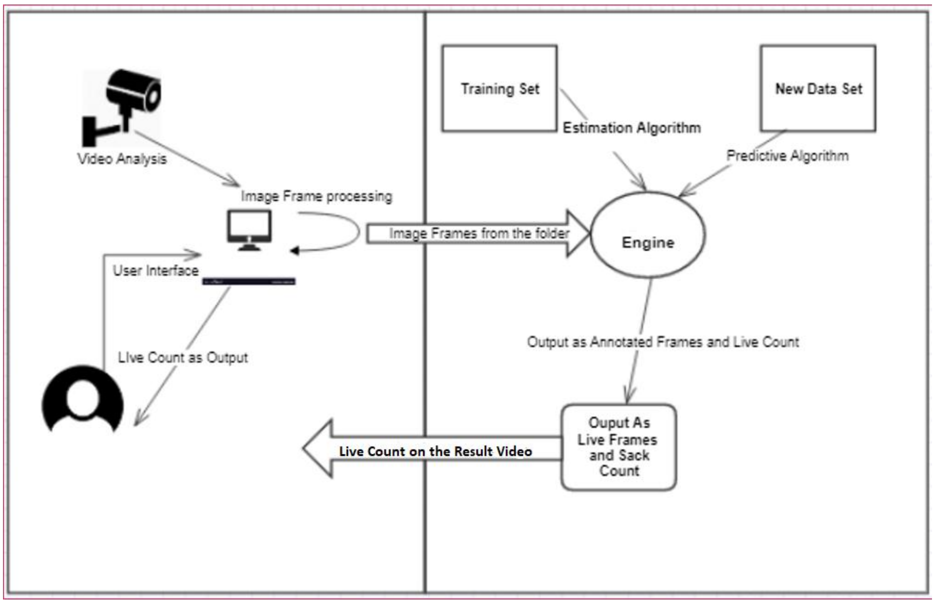

Fig1: Architecture diagram of Sack Count system

The goal of the Sack Counting System is to process the image frames passed to it via the image processing algorithm mentioned in the above diagram. The engine is trained as such, that, it is processes the image frames found in the frame folder and predicts the sack using a predictive algorithm and returns the live count of the sacks loaded on to the truck. The engine used is the OpenCV and the image frames are being fetched from the live video recording from the CCTV camera recording the video. The external interface of this engine begins with the collection of frames saved in a .csv folder. The frames are extracted through a live video stream of Camera installed at the work area. Then the OpenCV engine will take the frames one by one from the folder and start analyzing through a command-line interface. The engine will identify the correct frames and increment the counter, now the frames become annotated. Finally, the counter and annotated frames, the desired result will be the output video. The annotated result video will become the output.

The internal interface will be decomposed into two parts:Train and Test.

The Train part will be used to train the model for detecting the frame with a sack. Initially making of the dataset is done by own and divided randomly into training and validation set. It contains m images frames along with the label of sack marked on that frame using LabelImg annotation. Here, the engine will call the main function which in turn calls the train function which first initialises the parameters randomly and updates weights at each iteration upto some minimum cost. The model hyperparameters according to the variance and bias provided by the model are tuned. It is a continuous process to make a model robust to new images and provide more accuracy. This process will train the model and final parameters can be used to predict the new frame. In the test part or operational phase , the main function will be called, which refers to predict function. The predict function will check the image frame according to the trained model parameters. If the sack is present in an image frame, then the result would be there, in the form of an output result video which will show the number of sacks loaded/unloaded in truck and time and date of the loading/unloading in truck.

3.1 IMPLEMENTATION:

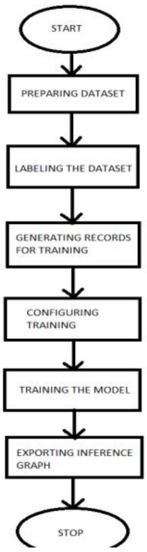

Fig 2: Workflow diagram

TensorFlow API[4] needs a huge bunch of images in hundreds or thousands in number of objects to train and test a good object detection classifier, best would be at least 1000 pictures for one object. To train a classifier for object detection, the training images should have random objects in the image which are not desired and considered as noise along with the desired objects, and should have a variety of backgrounds and different contrast conditions. There should be some images where the desired object is partially visible, overlapped with something else, or only half visible in the picture.

These are the basic steps required for the implementation of the Sack Count system:

- Create an image dataset as by collecting images that contain the desired object. The minimum amount would be about 100 images, ideally more like 1000 images or more.
- Annotate or label the images using LabelImg. This process is basically drawing boxes around the objects in an image. The labelling tool will create an XML file that describes the objects in the pictures.
- Split this data into train/test samples. Training data should be around 80% of the whole dataset and validation or testing set can be around 20% of the dataset.
- Generate TF(Tensorflow) Records from these splits.
- Setup a .config file for the Faster RCNN model.
- Train the model.
- Export inference graph from new trained model.
- Detect custom objects in real time i.e Test the model.

4. RESULT:

Image is taken as input from open images of sack which is available online or video and images obtained by digital camera can also be taken as input. Image frames should be extracted using the video feed from the CCTV camera. The object detection is done by applying the Faster RCNN[5] algorithm to determine whether a sack is present in the image or not. Results obtained is as shown below:

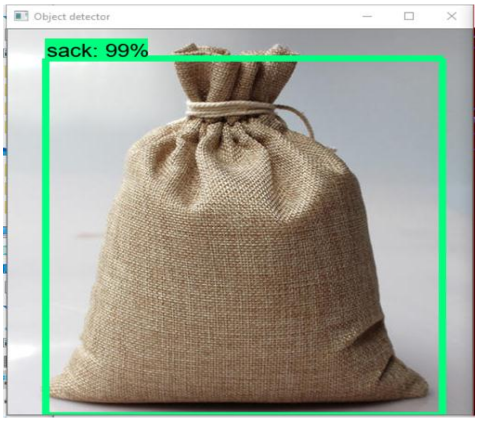

Fig 3: Test Result

5. CONCLUSION:

The paper concludes the latest development of computer vision using OpenCv platform and tensorflow api mainly in the logistics industry. An image-processing method is used to correctly identify each sack, in the warehouse using a conventional CCTV camera and object detection algorithm. For detection of sack in an image, dataset creation, image processing which includes object recognition, image features extraction and image feature classification followed by image detection and counting of sack is done within 20 sec as soon as input is given. The developed method shows the count of sack on the console screen and confidence score on the output screen.

REFERENCES:

[1] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y & Guadarrama S (2016) Speed/accuracy trade-offs for modern convolutional object detectors. arXiv preprint arXiv:1611.10012 .

[2] Redmon J, Divvala S, Girshick R & Farhadi A (2016) You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. : 779-788.

[3] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C & Berg AC (2016) Ssd: Single shot multibox detector. European conference on computer vision. , Springer: 21-37

[4]https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html

[5]Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" january 2016.