IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TOOL TRACKER

[1]Smita Unnikrishnan, [2]Alen Johnson, [3]Darvish Davis, [4]Anandukumar K.C
[1]Assistant Professor [2]Student [3]Student [4]Student
[1234]Computer Science Engineering Department
[1234]Universal Engineering College, Thrissur, India

Abstract: Object detection refers to the ability of models to locate and identify objects present in images. The existing system performs concealed object detection that identifies objects which are visually embedded in the background. An adverse condition faced by daily wage workers is losing their tools. A solution for this is to keep track or identify such tools. This is achieved by YOLO model. It is an algorithm that uses neural networks to provide real-time object detection and is popular because of its speed and accuracy. It provides a mean average precision of 87.52 % with better identification of tools. This system also explores chances for incorporating it into smartphones.

***Index Terms* – YOLO (You Only Look Once), CNN (Convolutional Neural Network), CV (Computer Vision), COD (Concealed Object Detection), ML (Machine Learning)**

I. INTRODUCTION

Imagine that our laptop is in front with massive eyes, and with all special sense organs. Not only that, it speaks to us about what they see. Is this can be extremely-modern revel in? Just don't get excited; computer systems can see. Object detection is a component hired in computer vision and image processing that offers with detecting times of gadgets in images and films. Well-researched domain names of object detection encompass face detection and pedestrian detection using video surveillance. From the mindset of engineering, computer vision seeks to recognize and automate obligations that the humans can do. Its responsibilities encompass techniques for obtaining, processing, reading, and understanding digital pictures, and extraction of excessive-dimensional statistics from worldwide to deliver numerical or symbolic facts, substantially in the form of decisions. Computers observe similarly to what a human does. If AI lets in computer systems to anticipate and make choices, computer vision gives it an ability to see and thus recognize. Computer vision trains machines to carry out those functions, however it has to do it in a whole lot less time with statistics and algorithms in preference to the optic nerves and visible cortex of a human eye. Nowadays deep learning techniques are typically used for those sorts of responsibilities. Its programs stretches to the sector of medicine, agriculture, transportation, and lots of others.

The proposed system makes use of the YOLO model for the monitoring motive. The modern model is YOLO V4. YOLO uses convolutional neural networks to allow real-time item detection as this is the want of the hour. Object detection in YOLO is performed as a regression and detects the snap shots at its very last levels. YOLO V4 is an item detector that may be trained on a GPU with a smaller mini-batch length. Tools are the primary approach through which human control their surroundings. A tool is an instrument to make adjustments on gadgets. Such tools are unavoidable for every worker. If such tools get lost, what will be their mindset? They will experience pain. There rises a hassle. If there can be a hassle, there might be answers for it. We all had come upon a vintage idiom i.e if there is a will, there can be a manner. It means that, if someone desires to perform an element, they will discover a way to do it, despite the fact that there are things that make it hard to do. But however there exists a question. Why didn't honestly everybody attempt to clear up this yet? Ok. Just depart it. Let us focus on how the answer can be. The tool tracker can remedy this. Most of those device are typically small in length and vulnerable to get out of vicinity within whilst at some stages. For this cause, managers or personnel themselves want to increase a streamlined method to track such property. An automated tool monitoring gadget is a notable solution and without this employees may lose their tools. This now not simply hurts your financials but additionally taints your credibility. With tools continuously being out of sight, it not only extremely delays your day by day workflows but also creates frustration. The proposed machine recognizes the absence of tool from its collection. A model is learned with all feasible snap shots of numerous sorts of device in the beginning using the YOLO algorithm. The found out version is only adopted for similar use if it successfully predicts the equipment. The model proposed in [2] plays concealed item detection using SI Net. This paper becomes the reference for the proposed system. Concealed is equal to 'have hidden'. It is difficult to distinguish the objects decided in such pictures. This hassle is solved using computer vision. But it has an excessive computation time and inefficiency in figuring out devices present in a photograph. Computation time is increased due to the usage of their very own neural network i.e, the search identification network which however is sufficient. But it makes use of a convolutional neural network within the search segment to process the picture and the identification phase employs several refinement blocks which are of same type. For the higher working of the model, snap shots of different types of toolkits are captured and saved already. Whenever the person wants to easy the doubts on the tools that she or he holds, they can capture a photo that consists of all their equipment. The model takes that photo, techniques it, extracts the records, and compares it with that of already captured toolkit pictures. If now not matched, the absent tools inside the second image are mentioned. Therefore they can check for it.

II. RELATED WORKS

The related works are arranged in the order in which technology evolved.

The strategy proposed in [13] is a gridlock control instrument to identify clog and keep away it in the metropolitan region. It looks forward mainly on 2 methodologies, for example, gridlock location and removal to direct the traffic stream. It works in light of the sliding window idea and hypothesis of a low thickness relationship. It is to decrease the time delay during a clog control. Initial one depends on the relative speed of vehicles and the other one depends on the sliding window idea. A code-based gridlock evasion technique is utilized here to clear the traffic. Traffic Congestion Detection(TCD) moves toward each vehicle on street, decides its relative speed by utilizing one-hop neighbors, and detects gridlock. In Traffic Congestion Avoidance (TCA), once the clog is recognized; the RSU tracks down an elective way to manage that. The TCA approach can be made sense of by thinking about a four-cross street idea. Every way has two paths one for ahead and one more for descending. The more the thickness, the more will be an ideal opportunity to stay away from the clog. The structure is assessed on the mean chance to recognize blockage and stay away from clogs under traffic streams and the number of vehicles in various situations. This approach is carried out just in a particular situation however it is important to consider all chances of specific street traffic conditions that might emerge in the future. The outcome shows that this system gives a total answer for traffic-related issues in metropolitan regions.

This framework [4] portrays the execution of a 0.435 THz imaging framework with a 1 mm-scale goal, which can distinguish objects that are visually embedded in the background extraordinarily. The functioning recurrence is certainly in the transmission band A, and that implies that the THz frequency that had been picked here is extremely encouraging later on significant distance remote detecting strategy. The pyroelectric detector, LiTaO3 crystal, is utilized for the enlistment of adjusted electromagnetic radiation in the THz recurrence range. Since the identifier just answers changes in the input THz power, a chopper is put just after the THz source to direct the information sign of this detector. Additionally, a channel is planned explicitly for this reason to lessen indicator that is sensitive to Mid-IR and noticeable light. The information obtaining unit block is intended for handling a signal from the pyroelectric indicator. It gives control of the amplitude modulator, synchronous detection, digitization, making of an information cluster from the indicator, and correspondence with the PC through the USB port. A lens is utilized to coordinate the THz wave onto the article. And afterward, another lens will divert the THz wave conveying the data about the item into the pyroelectric identifier. The item is put on a table for checking, constrained by a Controller associated with the PC. The filtering range and step time frame of the item can be changed by the software component in the PC. The location of this large number of components in the quasi optic way is very much intended to get a high resolution and distortion-free picture of the item to advance for testing.

This technique [14] proposes two modifications; one is about the exactness. For that, an additional convolutional layer is added to the network and named Fast RCNN Type 2. The other is connected with speed. For this to occur, the input channel is decreased from a three-channel contribution to one and named Fast RCNN Type 3. Fast RCNN Type 2 purposes every layer of Type 1 Fast RCNN yet with an extra third Convolution layer followed by Re Lu activation function layer which makes 13 layers. Fast RCNN Type 3 has 11 layers in total, same as that of Type 1 Fast RCNN yet with a little adjustment of information channel size; i.e rather than 3 channel input 1 channel has been utilized (input size = 32x32x1). The preparation choices utilized are SGDM advancement calculation with a momentum of 0.9, 1.e-06 beginning learning rate, 32 maximum epochs, and mini-batch with a size of 128. After the generation of ROIs by utilizing the edge boxes algorithm, the identifier is prepared by utilizing positive examples. Just those ROIs are considered by the organization for learning purposes whose boundary boxes have an IOU of 0.7 with the ground truth while the leftover ROIs which are marked as background are forgotten about during this task.

The framework proposed in [10] will permit the client to explore freely utilizing real-time object detection and identification procedures. This comprises wearable goggles along with headphones, IOT inserted platform utilizing Raspberry Pi 4. It has numerous sections that incorporate picture assortment, picture handling, object detection, expression acknowledgment, real-time ID, text to discourse changes, and a navigation system. The whole framework depends on goggles, which incorporate a camera and headphones. The working of the whole framework relies upon the picture gathered by the camera module. Then, at that point, the gathered pictures are handled utilizing Tensor Flow to distinguish various sorts of items present in that pictures. There are four methods of activity after getting the picture information. Describe mode, announcement mode, search operation mode, and the last mode. In depict mode, the framework utilizes nearby or cloud handling to clear up the client about the picture. The announcement mode is utilized by the framework to portray the article in the pictures progressively. In search activity mode, looking for objects and finding of directions are done by utilizing voice acknowledgment. In this framework, to recognize the article's definite position and distance from the client, the framework works out the angle from the client's area. In the last mode, real-time identification and expression acknowledgment are made by this framework. It utilizes a novel algorithm to identify new faces and save them on the server for future reference. After these steps, the framework creates text and interacts with it for the text to discourse transformation module. Then, at that point, the discourse yield is shipped off to the client through the bone conduction headphone related to the goggles.

The system specified in [20] uses faster CNN. It shares convolutional features of the entire image with a detection network. When compared to previous work, faster RCNN employs a region proposal network and does not require any external method for candidate region proposal. It tries to combine the different components of object detection into a single neural network. This neural network uses features abstracted from the entire image to predict each anchor. Also, it predicts all bounding boxes for an image simultaneously, which means the network cares about the full image and all of the objects in that image as well. This model tries to implement a CNN and then evaluate it on the Pascal VOC recognition dataset. The first convolutional layer of this network extracts the features from the selected image while the fully connected layers predict the output probabilities and coordinates. Faster RCNN uses a method called region proposal network to generate candidate

Regions Of Interest (ROIs), based on the input of the selected image. This is the opposite in fast RCNN which requires region proposals to be provided by an external source to make the same effect. The RPN is essentially built up by 3 convolution layers and 1 proposal layer. During training, the faster RCNN requires two additional layers; the anchor target layer and the proposal target layer. The anchor target layer is used to generate the target values. The proposal target layer is employed to generate the target class labels for the ROIs. The target regression coefficient of each class is used in the loss function of detection for the final detector. To take the most from this, the following four training attempts have been tried. Alternate string, approximately joint training, non-approximately joint training, 4-step alternating training. During the evaluation, there is only the need for a proposal layer. Target layers are available in Python. Therefore training faster RCNN has to be done using the python API.

The motivation behind the work proposed in [15] is to build a model utilizing a deep convolutional neural network and target discovery design (DCNN + Faster RCNN) to distinguish sound and affected tomato fruit. To acquire the large identification rate of tomato infection, the feature extraction network should precisely extricate the attributes of the picture of the unhealthy tomato. As a deep learning model, the convolutional neural network has the capacity for levelled learning and performs well in the extraction of characteristics. Here, three deep convolutional neural networks were chosen. To be specific, VGG16, ResNet50 and ResNet101. These three networks are particularly remarkable in picture feature extraction and have been generally utilized from the time of their development. They all were joined with the Faster RCNN structure to recognize tomato diseases. The general course of DCNN + Faster R-CNN is partitioned into four stages. First, come to the DCNN: In this model, a deep convolutional neural network is utilized to extricate the feature maps of the input picture, which is utilized by the following RPN layer and fully connected layer. Second is the RPN (Region Proposal Networks): This network is mostly used to produce region proposals. Third, comes the ROI Pooling: Here, a feature map of fixed size is acquired by utilizing the proposals created by RPN and the last feature map got by DCNN, which is taken care of as a contribution to the fully connected layer at the back for target acknowledgment and location. Fourth is FC and Softmax: In this, the properly sized feature map was given into the fully connected layer, and a Softmax layer was utilized to explicitly group the information. During this, the bounding box regression activity was finished to get the specific location of the tomato in the picture.

This system [16] proposes a spillage discovery strategy to acknowledge programmed recognition in the initial case. Faster RCNN is taken here rather than people (workers) to deal with SF6 infrared thermography pictures. Faster RCNN is a progression in CNN that has been generally utilized in various useful circumstances including picture recognition and face acknowledgment. It recognizes spillage as well as imprints the area of spillage. In Faster RCNN learning region proposal network is used first to create a specific candidate region; generally like a rectangular shape. The result of RPN has 2 neural networks; one is for giving the applicant the previously mentioned rectangular region and the other is to decide regardless of whether there is spillage. With the sliding window approach, RPN extricates various elements within these spaces by utilizing rectangular shapes of various sizes and different positions more than once. Along these, the qualities of the target region are re-addressed into a low-dimensional vector. This is taken into 2 networks. One is the boundary regression network and the other is the boundary classification network. The first network yields the final position of 4 vectors in target shapes to decide the position of the spillage and the networks that classify the boundaries give the candidate region a score that is decided as spillage and checks whether that score is higher or lower than a specified value to decide if there is spillage or not.

This paper [17], based on a customized Faster-RCNN model, develops a machine-learning system that can automatically detect parasites in thick blood smear images by employing smartphones. This paper develops a rapid and stable system for automated malaria diagnosis on smartphones in thick blood smears. To improve the detection performance of Faster-RCNN for small objects, images are split from a resolution of 4032×3024×3 pixels into regions of 252×189×3 pixels, and then a custom Faster-RCNN model is trained with the regions pointed here and corresponding ground-truth annotations that are done manually. The customized Faster-RCNN model includes four convolutional layers and two max-pooling layers. For testing purposes, blood smear images are divided into regions of 252×189×3 pixels in the same way that was done before the training phase, which is screened for parasites using the cascaded Faster-RCNN model proposed in this paper. The detected parasite coordinates are then again projected on the original image space for visualization, and evaluation and are also reserved for future use.

The proposed system defined in [18] is probably investigating the effect of using RGB, CIEXYZ, and CIELAB colour space on night-time human detection. To remedy the hassle associated with small-item size, the KAIST, cheap subset is used in which the annotations have a pixel peak of 55 pixels or extra. To have better bounding box annotations, the one and only one class used is the ‘person’ considering that the alternative classes have bad annotations. After filtering, pre-processing is completed on the dataset for removing noise and lacking values and as a result to make it robust. The first Pre-processing step for pictures is to up-sample the photo by 2. This is carried out because the duration of the photograph within the KAIST multispectral pedestrian dataset is only 640 X 512 pixels, it's too small to come across small objects. The second pre-processing step is to convert the RGB shade area to CIEXYZ or CIELAB. Open CV is used to up-sample the images and to exchange the coloration space. Then Faster RCNN hyper parameters had to be set. The Faster RCNN base network used on this is the Resnet101. The motive to choose Resnet101 is that it has higher overall performance in assessment than Resnet50 or VGG16. Resnet101 uses pre knowledgeable weights learned by the use of RGB photographs from the Image Net dataset as it's far a transfer learnable method. To enhance the overall performance, the Faster RCNN anchor generation had been changed. To adjust the anchor generation, the feature stride, anchor scale, and anchor ratio parameters can be changed. Anchor scales are utilized by the Faster RCNN to generate anchor boxes of diverse sizes. Anchor ratios are utilized by Faster RCNN to generate anchor boxes with diverse peak to width ratios. It is vital for the overall performance of Faster RCNN for the purpose that anchors with custom-designed height-width ratios suit some gadgets higher than pre-defined anchor packing containers. The evaluation metrics used right here are the mean average precision (mAP) metrics and log-average miss rate (LAMR). For learning the Faster RCNN human detection network, the proposed device follows Li’s experiment. All experiments will use a learning strategy with a speed of 0.001 for the initial four epochs and a learning speed of 0.0001 will be used afterward. The Faster RCNN human detection network might be completely trained for 6 epochs.

This research [7] had been conveyed to give a safe environment to drivers and visually impaired people. This paper illustrates tests led on side of the road traffic symbols to increase the efficiency and accuracy. Two algorithms; faster RCNN, and a single shot multi-box algorithm are tested here. The candidate object's localities are proposed by region proposition networks alongside the VGG-16 locator which then, at that instant, performs image classification at the top position. The bounding box proposals are anticipated by the layers that lie between them. In the following stage, box proposals are utilized to crop features, from the intermediate feature map. These maps are fed as input to the remaining feature extractor block for anticipating the class labels and for refinement purposes for every proposal bounding box. A single feed-forward CNN identification framework that predicts classes alongside anchor offsets directly without requirements of second stage classification is utilized here. This framework utilizes a multi-scale convolutional bounding box. These outputs of the bounding box are connected at the top to the feature maps that are multiple in quantity.

This exploration of [1] contributes to an endeavor to look at the utilization of item recognition and instance segmentation for crisis vehicle discovery (rescue vehicles). Faster RCNN and Mask RCNN are utilized for this interaction. For the substance detection to be done, the guideline of transfer learning is utilized here. i.e these base networks are already learned on enormous datasets to limit the number of calculations. They are likewise made completely convolutional to include the contributions of various aspects. Furthermore, these base networks are provided by object identification networks like Faster RCNN, Single Shot Detectors (SSD), and Region-based Fully Convolution Networks (R-FCN). The CNN model that has been decided for execution is Faster RCNN with a Res Net spine. The instant segmentation fundamentally utilizes the utilization of an adaptable system called Mask RCNN. It has two chief stages. The principal stage expects the presence of the article in some locality of the input picture. It is known as the Region of Interest (ROI). The subsequent stage figures the likelihood shows Intersection over Union (IOU) bounding box and the cover around the picture in light of the outcomes from the primary stage. The two phases are moulded in the spine. This network has three parts in particular FPN, RPN, and Backbone network architecture. Here the bottom-up design of FPN is executed for characteristics extraction from the feed-forward network. The RPN is a light and straightforward network that checks the FPN bottom-up and proposes plausible regions in the picture where the item is probably going to be available. It then perceives a few different areas by fitting various bounding boxes as per the specific IoU values. The spine is a multilayered neural network that includes feature maps of the input feed-forward network. Here ResNet50 is utilized. Since it's a shallow type of design. Fine-tuning assists the model to achieve higher precision with less computation time. The Stochastic Gradient Descent algorithm with a small mini-batch is utilized to refresh the loads and momentum which in turn limits the loss and assists to reach a most exact worth.

The system proposed in [3] will give a novel technique to classify and detect the type of teeth (viz., incisors, molar, premolar, canine) and further a few fundamental oral anomalies that include fixed incomplete dental replacement and affected teeth. A python tool for marking the ideal tooth in the pictures is utilized. This is done by drawing a square shape around every single tooth and giving it a label. Then produce TF records utilizing the labelled images which will act as input data to the Tensor flow training model. Then convolution is applied to the previously generated TF records. In that, the convolutional layer is the first layer that is utilized to extract high features from given input images. It is also an element-wise multiplication and generates an output from this multiplication i.e feature maps. The feature maps are subjected to an activation function known as Rectified Linear Unit to increment non-linearity input images. Then, at that point, the image input is given to max-pooling layers to reduce the dimensionality of the image and finally given to the fully connected network which utilizes a soft-max activation function to characterize and recognize various teeth. In the Faster R-CNN architecture, a Region Proposal Network consists of a convolutional neural network, a bunch of anchors, and a Region Proposal Layer. On the last convolutional layer, a sliding window of size n × n runs on the feature convolutional map set. A set of 9 anchors are generated for every window with the same centre x_a, y_a with 3 different aspect ratios and scales. Anchors are different-sized boxes that help the developer to detect objects of different sizes. To assess an object detector, ground-truth bounding boxes and predicted bounding boxes are employed. The extracted features (n × n spatial) from the feature maps of convolutional layers will act as inputs to the other two networks which are dedicated for classification and regression respectively. The regression produces a predicted bounding box(x, y, w, h). The classification output gives a probability P which depicts the contents of the predicted box.

A method in this paper [8] provided a strategy to Traffic tracking and assessment in toll plazas in India. The proposed answer is a mixture of models knowledgeable to accomplish objectives like automobile identification on stay video streams, Vehicle computing, Vehicle version type categorization, and automobile registration number identification. The challenge started with the identification and localization of several kinds of automobiles and shape discriminating borderline for the located devices into video frames and feed on the identified gadgets to be collected locally and used for further making ready in outlaid milestones. In object detection, we smash down the real photo into several snapshots and execute the item localization on everything. To discern the issue of categorization and regression, the gadget has various models which use several outlooks alongside R-CNN to filter out the ROIs. Fast-RCNN filters out the ROIs from Feature Map. Faster R-CNN makes use of the Region Proposal Network (RPN) to filter the ROIs and SSD. For object recognition we used tensor flow object recognition APIs; one of the strongest equipment to accomplish the objective reputation milestone and use a set of rules for implementation. Input image implemented with layers of convolution layers (with RELU activation function). Max pooling layer to compact the defined characteristic maps using filtering out the max values of the characteristic map values to interest on the principal abilities of diagnosed gadgets. The output of the pooling layer demolished the feature map; then float through fully connected neural network layers. Soft-Max activation allows doing multi-class categorization. For the diagnosed automobiles in several video frames by using Faster RCNN implementation it became a challenge to study the moving vehicles and as a result, we implemented a naming common experience to residence the identified items and look at them in the next frames. An interplay of location directory, a counter to tune a variety of feed on images, class index identification, index name, and object identity had been used as necessities to form a call for the diagnosed cars. A particular object identity mentioned is described through the Centroid tracking algorithm are given inside the defined call location it is then kept in CSV fashion and subsequently combined for detailed analysis. Such as computing amount of automobiles. Further to the amalgamation in the answer the recognized vehicles with bounding boxes are fed on and used for car type and vehicle registration number recognition. We used VGG16 based picture classifier for detecting the recognized cars in the first stage of our device. The identical feed on the photo of object recognition level of our system was used as an input for automobile registration number identification. Background subtraction was accomplished with Otsu thresholding. Background subtracted snapshots were then threatened to decomposition in which each character was further forecasted with an SVM-based character detection algorithm. Finally, the output of a majority of these fashions is mixed to form a CSV document with penetrates like automobile counts, car registration number, and vehicle type classification including timestamp revealing the date and time. This perception then diverts over the dashboard for stay gazing and in addition evaluation.

This version [19] used SSD (single shot detector) algorithm and mobile net method to execute the recognition and classification. Localization determines the coordinates of the identified sample and their corresponding borderline. This paper deduces a manner of object recognition and kind on a platform in which the processing energy isn't always strong. Here the model is trained on the MS COCO dataset that is contemplated to be one of the first-rate datasets for executing object recognition. The method used tensor flow deep learning API as its backend. Using the power of CNN, the identification accuracy is incrementing in parallel with the classification accuracy and now not through decreasing the speed. A videotape from a drone can be used as input for undertaking the outputs. This version executes an object-based reputation tool that is prepared to perform excessive accuracy and run at 5-7 frames per second. COCO datasets used in this method consist of numerous varieties of devices. Mobile net is being used as it is suitable where computational power is small. The pictures have been resized to 224 * 224 to lower the price of the performance. Various filters are applied to the resized photos to disregard the noise from the pictures. Then, the pictures are used for training and testing. The model uses SSD architecture that has a small 3*3 convolutional layer for feature extraction.

This model used in [21] is DSP board embest devkit 8500D. The Microsoft life cam HD300 is utilized in this model. Angstrom operating system installed in this and it is programmed in c ++ language combined along with open CV library. Frame differencing technique is utilized herein in which consecutive frames are decremented from both to delete the stationary background. Background diminishment is a method for detecting any change in a given frame by reducing settled frames from fixed reference frames. The DSP board uses its floating and MAC (multiplication accumulation units) processes to improve its adequacy. This model (DSP) depends on TI DM3730/AM3715 processor integrating ARM cortex A8 part at 1GHz and DSP comes at 800MHz. DSP board is a portable model with a Linux angstrom installed operating system. It goes with the inherent GCC compiler which can organize C and C++ programs. What's more, is the open CV library which speeds up the image processing applications. The DM3730 central processor is being utilized in the DSP. The programmable DSP helps in figuring out different signal processing tasks like processing and analysis of digital images, digital filtering, and math functions. The DM3730 processor can comparatively be utilized in large definition video applications which consist of processing a high amount of data. Six general purpose input/ output (GPIO) banks are there in it. Each GPIO bank handles determined general purpose pins. The display subsystem gives the logic to show a video outline from the memory frame buffer (either SDRAM or SRAM) on a Liquid Crystal Display (LCD) board that has been appended to the unit. The working design utilized in this unit combines four sections xloader, vboost, kernal, rootfs. The operating system helps various social events with cherishing open CV library, python and java script, web applications, and many others. It is predominantly utilized in this model as it has a better effect on the open CV libraries and what's more, is that the USB camera can be used with great easiness.

This version mentioned in [12] advises an implementation for recognizing the identity and location of objects in the real-time video preview and coinciding 3-D graphics on them in IOS applications. The methodology for combining the YOLO model with Augmented Reality involved the following steps; practicing a tiny YOLO ML version, passing frames to the pre-trained types to request output, handling outputs to draw out predictions, performing hit test to get the 3-D role of an object, and covering 3-D graphics at the placement. To prepare the tiny YOLO ML version, the Turicreate engine and the INRIA annotations for Graz-02(IG02) dataset were used. A stop flag becomes set up to inform the utility whilst to prevent from trying to pick out gadgets. The flag was set to false at the beginning. It could be alternately set to false if only a specific number of occurrences of an item had been to be identified. An outlook for displaying AR experiences that boosts up the digital camera outlook with 3-D Scene kit content turned into a feature that hired the whole display. The function might continuously call others in its body and then calls itself if the forestall flag turned false. The function would seize a frame using the screenshot API. The image turned into some other picture after passing through certain functions. In the classification request, the Threshold (Non-Maximum Suppression Threshold) is set to 0.5. The tiny YOLO version is liable to forecast several comparable predictions joined with a single item. i.e single object instance. In post-processing, any forecasting with Intersection over Union with the largest confidence forecasting small than the brink is removed to avoid several predictions joined with a single object instance. The output had been dealt with successive constituents.

The major reason for this model [6] is to bring together the more accurate object recognition method for Augmented Reality the use of communication between the deep analysing treating and the micro mild holo lens as input or output device. Here the version describes the number of techniques for item identification by the usage of YOLO algorithms through a holo lens. The software program development donation is based totally on the purchaser-server interaction connection. For the hardware material, it used two Graphics Processing Units (GPU) of NVIDIA kind Quadro P4000 to compute the computational time of the server factor; usually the handling of YOLO algorithms. A Microsoft holo lens version 1 was used for the client-side to attain its digital camera as an entering arena to item recognition in the real world. This digital cam included at the front of the system permits programs to appear what the man or woman sees. Developers have a technique to manage the virtual camera really as they do for colour cameras on mobile phones, portables, or computer systems. In this model, holo Toolkit; the library of micro tender shipped in to govern gaze, gesture, and cameras. The server-facet is composed of the darknet library of Redmon to run the YOLO algorithms that are carried out in C and CUDA programming languages.

The research in [5] goals to suggest a more comprehensive solution. Item recognition using YOLO gets input from the camera unit, the input from the unit will be processed using weights attained from the learning program to provide a model that can identify the presence of cigarette items, the outputs of the item identification program using YOLO in the form of integer value 1 or 0 which will be done later. OR logic gate operation with outputs attained from smoke identification programs using smoke sensors. In this procedure of identifying smoke by a smoke sensor, using the input of smoke circulating in the air in public facilities, the level of CO gas pollution as the dominant gas considered in cigarette smoke is 50 Ppm, if a CO gas of larger than 50 Ppm is identified it will result in an integer value of 1 and if not it will problem an integer of 0 which will then be carried out the OR gate logic procedure with item identification outputs using YOLO. The result of the model as a whole is the public service announcements about the trouble of smoking and also caution using buzzers triggered by the result bit of the OR operation between item identification using YOLO and smoke identification using sensor readings. In some conditions, where the result of each identification unit is zero (0), the result bit of the OR operation is also zero. The model will display a random PSA with various topics, that rely on the campaign list that is dispelled as the default list of the display. That way, the model is producing the educative PSAs using the pulsing method as previously described.

This method [9] provides a UWB imaging system, consisting of a modified rotating UWB antenna array, RF circuits, and the 2D implementation of the late-and-add imaging algorithm. In the previous UWB imaging method, there are four accepting antenna factors placed side-by-side as a direct accepting arm at a distance away from the solo transmitting antenna to form the antenna array. It is called that if incrementing the accepting antennas in the array, the radiated beam-width of the antenna array will be upgraded, which will better the cross-resolution of the imaging. In this case, based on the present UWB imaging method, two more antenna factors are added to the rotating antenna array, as well as reducing the distance between the transmitting antenna and the accepting antenna. While more antenna factors are added to the array and the accepting antennas are close to each other, the mutual coupling between the antennas is bottom -15dB during the operating frequency band. during the six RX factors, the array will have a narrower radiation beam compared with the four RX factors. The 2D reconstructed picture is the interaction of each rotation plane of the antenna array. In each plane, the model is based on the late-and-add algorithm. Here X-axis is imagined as the direction along the width of the aim while Y-axis is along the height of the aim and the Z-axis is along with the below range distance. In the XY plane, when accepting antenna arrays A0, B0, C0, D0, E0, and F0 are at the angle of 00, the re implemented picture can be accomplished. When rotating the antenna array to various angles, the coordinate X-axis and Y-axis will be settled to rotate at the same angle with the accepting antenna array so that the re implemented picture will be rotated in terms of the varied coordinate, which is the same as the area. Then rotating a circle, the area highlighted will be combined into the final re implemented output.

This strategy in [11] proposes another imaging model by de-aliasing. Here the proposed framework processes mathematical re-enactments to extend the presentation of the imaging model proposed. In the re-enactment, we pick a plane-type transceiver antenna with a level of 2m and a width of 1m. The MMW frequency band is set to that from 27.52GHz to 30GHz, where 32 frequency points with the spacing of 80MHz are taken in the re-enactments. For correlation, two settings on the separating of the array factors are comprised, one is 2mm for the non-aliasing case and the other is 5mm for the aliasing case. Numerical tests are figured for imaging two sorts of targets, one is a plate with sticks, and the other is a pattern with three-dimensional dispersing points. In experiment 1, the plate is taken where seven packs of sticks are set vertically and horizontally and the plate is corresponding to the plane of the exhibit with a distance of 50cm. Each set comprises four equivalent width sticks, and both the stick width. The separation between the sticks ranges from 1mm to 7mm. For effectiveness, here four frequency focuses, i.e., 29.76, 29.84, 29.92, and 30GHz, are used. We can separate the sticks into sets of 3 and 4mm in width. Additionally in both the vertical and horizontal directions. 4mm width sticks won't be recognized, while the imaging yielded by our proposed de-aliasing model, is practically comparable to a conventional methodology both in resolution and lucidity. In the second experiment, the example with three-dimensional disperse is fit on planes with z's = 30, 31, 32, 33, 34 cm. It has been observed that by utilizing the conventional model the imaging dissipating points inside the circle are unclear, while the imaging with this de-associating model gives clear dispersing pointed pictures, and the details where these are discernible.

III. EXISTING SYSTEM

Existing system [2] performs concealed object detection. Concealed means something hidden or kept secret. It is a cumbersome task to identify such objects with naked eye. To understand, consider a yellow coloured cat lying on yellow carpet. This type of pictures have very slight intensity variations among the objects present in them. But this can be easily differentiated by CV. For this to occur, the experimenters employs a SINet (Search Identification Network). They used the TEM in this system to incorporate more discriminative feature representations during the searching stage. In identification phase, to address the issue of ignorance of structural and textural details, a principle strategy is introduced to mine discriminative concealed regions by erasing objects. i.e the output reverse guidance is obtained via sigmoid activation function and reverse attention. Reverse attention is used for mining complementary regions and details by removing the existing target areas from output features at sides. Finally, the residual learning process, termed the GRA block is introduced, with the assistance of both the reverse guidance and group guidance operation. Due to the involvement of this specially designed network, the computational time is high. Not only that, it only detect single object from an image. To overcome this disadvantages, the following system has been developed.

IV. PROPOSED SYSTEM

The proposed system identifies missing tools from a toolset. This approach has become the proposed system only because of the impact that YOLO made in the field of computer vision.

4.1) Methodology

The methodology of this system is to track or identify tools using the YOLO algorithm. This section sketches the details and method that how the study that has been conducted. This is a web-based application that can be used anywhere and at any time. For that reason, all workers may use this. A web application is an application program that is stored on a remote server and delivered over the internet through a browsing interface using popular search engines like google, Mozilla fire-fox, etc. web services are web apps but not all. Methods to implement this on smartphones are on-going. This can be achieved using flutter technology. Flutter is an open-source framework for creating native mobile applications that is launched by google.

It allows developers to build mobile applications for both iOS and Android with a single codebase and programming language.

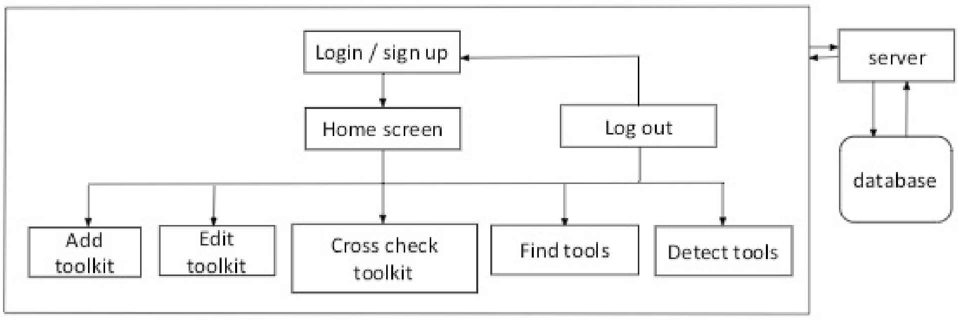


Figure 1: block diagram of tool tracker

Initially, the user can browse for the tool tracker on the internet and can make use of it by clicking the link. When they enter this site they can log in to this service by using their phone number and password on which they had built their tool tracker account. If not a user they can sign up by providing their name, phone number, and password. After this, users are directed to the index page. This page contains a navigation bar on which all the service provided by the tool tracker are listed including the three main modules; detect, find and crosscheck toolkit. From this page, the user can choose an option to proceed. After all, the user can sign out by using the sign-out option provided in the navigation bar. All these tasks are accessed through the server and the data are stored in the database for verification and future use. These are visualized in figure 1.

4.2) YOLO model

YOLO is an abbreviation for the term 'You Only Look Once' algorithm. YOLO proposes the use of a neural network that makes predictions of bounding boxes and class probabilities all at once. YOLO V4 is the 4th installment of YOLO. It was invented in 2020 by Alexey Bochkovsky. It achieved state-of-the-art performance on the COCO dataset which consists of 80 different object classes. It is a one-stage detector. With the release of this version, there is a significant increase in the inference time of the model. The architecture of YOLO comprises of backbone, neck, and head. Backbone is to extract the needed features, the selection of the backbone is a key step as it improves the object detection. Neck is used to collect feature maps from different stages. Head is to perform dense prediction i.e coordinates of the predicted bounding box along with the confidence score of the prediction and the label in the case of one stage detector. The proposed system works with advantageous YOLO V4 due to its improvement in the mean average precision (mAP) by as much as 10% and the number of frames per second by 12%.

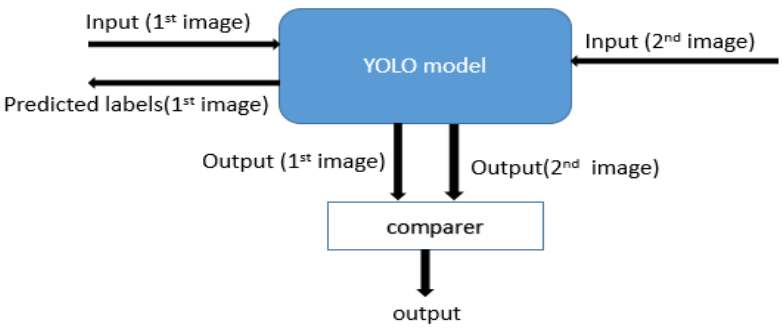


Figure 2: architecture of tool tracker

The crucial functionality of this system is to detect the missing tools. Here the trained model will do all the tasks of tool tracking. The architecture of the proposed system is included in figure 2. At first, the image on which the reference has to be made is processed using the YOLO model, and the details of the tools present in that image are displayed for confirmation purposes. Parallely the results are stored in the server for retrieving whenever cross-checking is required. This image will be mentioned as 1st image in the remaining sections of this paper. When the user needs to cross-check his/her tool types, they can use the same model (which has been used to process the 1st image) to process it by capturing a clear image of their all tools. This image will be mentioned as 2nd image. The output of both 1st and 2nd image received from the model will be fed as two inputs to the comparer module which compares the details of both images and display the details of tools that are absent in the 2nd image.

4.2.1) Dataset Collection

The dataset in almost all cases determines the accuracy of the model. Several other factors like the choice of the right configuration file for training, augmentation techniques...etc determine the accuracy level. For the validation of the proposed classification scheme, the dataset includes ten different types of tools: chisel, file, hammer, plier, saw, screwdriver, spirit level, tape measure, utility knife, and wrench. More than 200 images of each type of tool with high resolution and on different backgrounds like white and dark backgrounds were collected and contributed to the dataset. To improve the accuracy of the model, images including the different types and the same types of tools were also added. The dataset collection was a cumbersome task as no public datasets were available on any of the corners of the internet. So to achieve this everyone has to manually search for each tool on the internet and download it one by one. To make it as easy as possible, a google drive folder has been created so that all team members can collect the dataset and upload it or just drag and drop the images to it. Another problem faced during dataset collection was that the search on the internet doesn't provide accurate search results. To understand this problem, let us consider the scenario of searching for a tool; hammer. The search result includes hammer in white background, dark background, with the watermark on the needed portion of the image, unordered content in the background,...etc. Photographs with the watermark on the content cannot be included in the dataset as it leads to misclassification by the model during training. The images of tool 'file' were less in quantity as it is not available abundantly. Some data from all types of tools were captured using smartphones and high-resolution cameras for improving the precision of the model.

4.2.2) Data Preprocessing

Data preprocessing is a process of preparing the data in a way suitable for the machine learning model. While creating ML projects, it is a need to come across the clean and formatted data because real-world data generally contains noises, and missing values, and maybe in an unusable format that cannot be directly used for learning the models. So, the data preprocessing step is involved. It also increases the accuracy and efficiency of a machine learning model. In this project data preprocessing techniques includes changing the extensions and resizing images.

4.2.2.1) Changing extension

YOLO only supports images with a .jpg extension for training the model. So a python code has been written to change all image files with extensions other than .jpg to YOLO supported file format. The change in file extensions affects the dataset both positively and negatively. In positive cases, it accurately changes the extensions, and sometimes along with this the background also gets changed to black colour. Negatively some portions of the image get blurred or the pixels get unordered which makes the image unidentifiable.

4.2.2.2) Image resizing

The dataset includes image files with a resolution of varying degrees. This also includes square and rectangle shapes according to the width and height of the images. YOLO model requires all image data in the dataset to be resized to a specific resolution which can be chosen by clients. Examples are 416*416,512*512...etc. This is because the model learns by dividing the images into a n*n grid which is selected according to the values of width and height mentioned in the configuration file. Therefore resizing to a standard value is always necessary.

4.2.3) Data Augmentation

After the preprocessing, the datasets were augmented to improve the robustness of the proposed model before being input as training data. Among the augmentation techniques; flip, 90-degree rotation, saturation adjustment, brightness, and exposure level adjustment were adopted. These methods are selected as they improve the model performance in substantial ways, make the model insensitive to camera orientation, and more resilient to lighting and camera setting changes. After this process, the size of the dataset has been increased thrice i.e more than 500 images for each type of tool.

4.2.4) Annotation

Dataset annotation is an important step involved in CNN based model. YOLO is also a CNN-based model. The annotation format supported by the YOLO model is the bounding box. Each data is needed to be labeled manually. There are no shortcuts for this task. For that reason, we can conclude that the most tedious task in most ML projects is the annotation process. This format contains one text file per image containing the annotations and a label map that maps the numeric IDs to human-readable strings. The annotations are normalized to lie within the range of 0 and 1. Bounding box annotations has become quite popular as it follows the Darknet framework's implementation of the YOLO models. Each text file contains the class id, x-center, y-center, width of the image, and height of the image. The model learns from the coordinates in the text file where the object lies in the corresponding image file and to which name it belongs. Some care that all must take while annotating images for the object detection model is:

a) Create specific label names.
b) Create tight bounding boxes.
c) Annotate all objects of interest.
d) Annotate occluded objects.
e) Annotate the entirety of an object.

4.2.5)Model training

Here comes the heart of the proposed system i.e ML model training. Before this, the dataset verification has to be performed. This task includes listing of contents of all text files for each type of tool to verify all annotations are in YOLO format and don't contain any wrong annotations like missing any of the coordinates, width, and height of the image. This is also a difficult task. There is also a need to make sure that all images after augmentation and preprocessing are visually good enough to get properly learned by the model which determines the accuracy of the model. If any useless files are encountered their entry will be eliminated from the dataset. The fully cleaned dataset will be uploaded to google drive in a compressed format and is unzipped there. This is done to access the dataset easily for training. Model training is done on google colab with help of GPUs (Graphical Processing Unit) s. Co-lab is a hosted Jupyter notebook service that requires no setup to use while providing access free of charge to computing resources including GPUs. Some model training continues for hours and days. Not only that, everybody can't afford the price of GPUs. In such cases, online model training is a good option to proceed with. The type of GPU used for training in co-lab is Tesla K4 along with 16 Giga Byte of RAM and other requirements. It supports over 30 programming languages including Python, R, and even more. It means that it can run code written in all those languages. For training to be done, there is a need to mount the google drive with co-lab. After mounting, create a folder to store all the works that are going to be done for training purposes. In this folder download the darknet folder by cloning from GitHub. Darknet is an open-source framework written in CUDA and C languages. It is fast and supports GPU computation. This folder contains different configuration files, predictions, and results obtained during the testing of the YOLO model on the MS COCO dataset. Then download the makefile and make some changes to install YOLO prerequisites. This includes changing of values of some libraries and dependencies like CUDNN, GPU, Open MP, LIBSO, HALFCUDNN, and OPENCV to 1. CUDNN is a GPU-accelerated library employed for deep neural networks. It provides highly tuned implementations for forward and backward convolution, pooling, and activation layers. Open MP uses a portable model that gives programmers a simple and flexible interface for developing parallel applications for different platforms. Open CV is an open-source computer vision library that helps all users during computer vision based applications. After making changes in the makefile according to the need of user, the next step is to download the pre-trained weights - 'yolov4.weights' on which the other dataset had already been trained. This is done to include this weight file during training as the YOLO model is a transfer learning-based model. Transfer learning is a machine learning method where a pre-trained model is reused as the starting point for a model on a new and a similar task. For example, knowledge gained while learning to recognize a sparrow can also be applied when trying to recognize an eagle. Then split the entire dataset for training and testing using python code which results in the creation of a text file each for the train and test set containing the path to all image files as selected for training and testing. After all these steps we are at the doorstep of model training and start the training using a single line of command which includes the path to the configuration file, the repository where data had been stored, and the name of the weight file which will be used for training . Bootstrap 4 of HTML, CSS, and JavaScript languages were used for frontend development. The server-side scripting was done in the python flask framework. The database used here is the Maria DB database. This database is more user-friendly and the queries are more simple to handle than that of My SQL.

4.3)Modules

There are mainly three modules in this system.

4.3.1) Detect tool

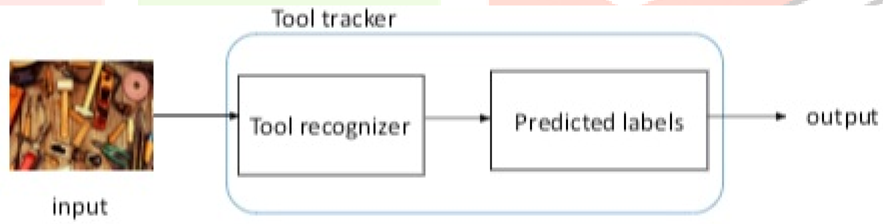


Figure 3: detect tool

This module is used to locate and identify the tools present in an image. The image may contain numerous different types of tools as a worker requires a large variety of tools for their job. Therefore to get such information, workers can use this model which reduces their task of manually identifying each tool and the details of each. This module takes only one input. In this module, the image of a tool on which detection has to be performed is fed as input to the YOLO model. The model predicts the output containing the details of tools including the type and count of each tool present in that image.

4.3.2) Find tool

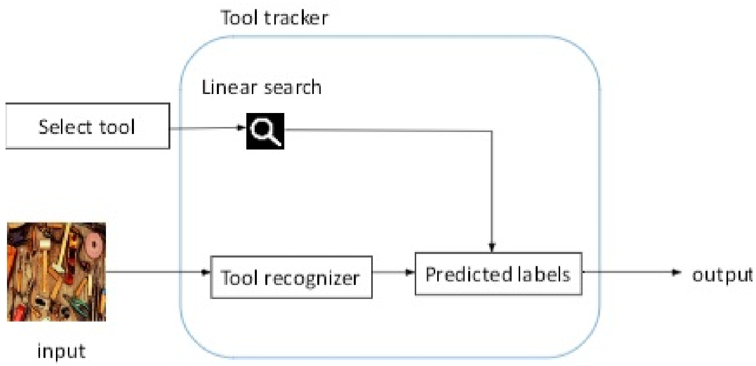


Figure 4: find tool

The find tool module is mainly implemented for verification; to verify if the specified tool is present or not. This module takes two inputs; one is the image input on which the search has to be performed. And the other is the text input; the name of the tool which has to be found. Initially, the image input is fed into the YOLO model to predict the type of tools present in the image. On the output generated by the model, a linear search is performed to verify whether the tool which the user needs to verify is present or not. This module helps the user to clarify his/her doubts about the availability of tools. Users can find this module on the navigation bar in the index page.

4.3.3) Crosscheck toolkit

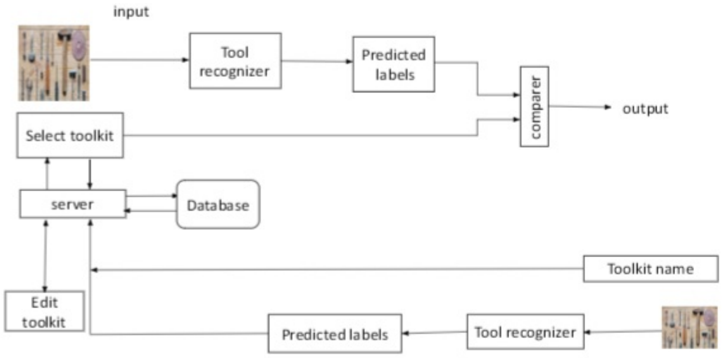


Figure 5: crosscheck toolkit

Crosscheck toolkit is an unavoidable component in the tool tracking system as it provides the details of missing tools in an image. A toolkit is an image containing all possible tools for a specific type of job. The aluminum fabricator toolkit is the clear image containing all available tools in that field of work. Similarly, the carpentry toolkit is the image containing all possible tools in that particular field of work. So there will be a different toolkit for each type of job. Here in this project, only the carpentry toolkit is considered due to less availability of resources like experience, time, teammates…etc. This module can be divided into two components:

4.3.3.1) Manipulation of toolkit

This component is used to make any changes to the toolset. It includes

4.3.3.1.1) Add toolkit

In this section, a new toolkit can be added. If there is a need to add more toolkits, provide an image of the new toolkit to model for processing. The processed image along with the result will be stored in the server under the name specified during the addition process.

4.3.3.1.2) Edit toolkit

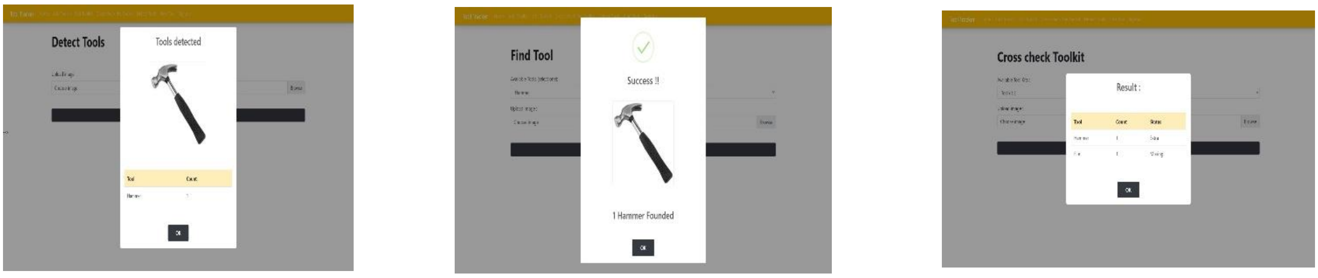
This feature enables the user to change the toolkit if needed, and delete the useless toolkit.

4.3.3.2) Crosschecking of the toolkit

This module also takes two inputs in a similar way to previously mentioned module but both of them are image inputs. When we select the type of toolkit, the stored result of that toolkit; i.e the predicted labels are fed as the first input of the comparer. When the second image on which test has to be done is captured, it is processed through the YOLO model and the model predicts the classes of tools present in it. This result is stored in the server as well as acts as the second input to the comparer. The comparer compares results provided at both input ports and generates the details of tools that are not present in the image provided at the second input port.

V. RESULTS AND DISCUSSION

Generally, YOLO achieves state-of-the-art results for real-time object detection and can run at a speed of 65 FPS on a V100 GPU. During training, at the 5422 nd iteration, YOLO model that employed here provides an accuracy of 87.52%. When comparing with the existing system, it provides better accuracy on 69 classes of different varieties of concealed objects. The model stores the weight files for every best accuracy in the backup folder. Similarly an updation is done at every hundredth iteration on other weight file with name 'last weights'. The last weights can be used to retrain the model if the model gets disconnected during training. The best weight plays a major role in this system. This will be used throughout the project. When a client uses this system for any of their requirement, they call this best weight file. The final output of each module is displayed in a tabular form containing the types and counts of each tool along with the input image. It is shown in the following figures.



(a) (b) (c)

Figure 6 : results achieved by (a)'detect tool', (b)'find tool', (c)'cross check toolkit' modules of tool tracker

Table 1 : comparison of result

System	Accuracy	Computational time	Detection
Existing System (COD)	80%	High	Single object from an image
Proposed System (Tool Tracker)	87.52%	Low	Multiple objects from an image

VI. CONCLUSION

In short, each one desires a tracking system to keep tune of their possession to prevent their losses. Higher the responsibility, the more will be the care. This assignment gives a carried out answer for tool loss and is a state-of-the-art project in the blue-collar job fields. YOLO is known for its predictable pace in real world packages. Instead of selecting the interesting part of the picture, it predicts types and bounding boxes for the whole photo in a single run of the algorithm. CNN and Open CV libraries have been used right here. Some limitations of this method consist of the dearth of tool types. This approach that the model use right here only specializes in the carpentry toolkit. So only misplaced device in that category may be observed. Another predicament is the quantity of modules included in this model. More capabilities can be introduced inside this, inclusive of stay tracking of lacking equipment and several others. In overall, YOLO model had made the task easier.

VII. ACKNOWLEDGEMENT

We have great pleasure in publishing paper on tool tracker. We take opportunity to express our sincere thanks towards our guide Mrs. Smita Unnikrishnan; Department of Computer Engineering, for providing the technical guidelines and suggestions regarding the line of work. We would like to express our gratitude towards her constant encouragement, support and guidelines through the development of the project. We thank Dr. Sreeraj . R; Head of Computer Science Department, for his encouragement during progress meetings and for providing guidelines to write this project. We thank Mrs. Chinju Poulose; project coordinator, Department of Computer, for being encouraging throughout the course and for guidance. Lastly, we would like to thank our college principal, Dr. Jose.K.Jacob; for providing lab facilities and permitting us to go on with our project. We would also like to thank our colleagues who helped us directly or indirectly during this project.

VIII. REFERENCE

[1]. Kaushik S, Abhishek Raman, Dr. Rajeswara Rao K.V.S "Leveraging Computer Vision For Emergency Vehicle Detection-Implementation And Analysis",11th ICCCNT IEEE, 2020 .

[2]. Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao, "Concealed Object Detection" , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , july 04,2021.

[3]. Anuradha Laishram, Khelchandra Thongam " Detection and Classification of Dental Pathologies using Faster-RCNN in Orthopantomogram Radiography Image", 2020 7th International Conference on Signal Processing and Integrated Networks.

[4]. Feifei Xin, Hongyan Su, Yong Xiao, "Terahertz Imaging System for Remote Sensing and Security Applications", 2014 3rd Asia-Pacific Conference on Antennas and Propagation.

[5]. Lathifah Arief, Alif Ziden Tantowi, Nefy Puteri Novani, Tri A. Sundara," Implementation of YOLO and Smoke Sensor for Automating Public Service Announcement of Cigarette's Hazard in Public Facilities", 2020 International Conference on Information Technology Systems and Innovation (ICITSI).

[6]. Haythem Bahri, David Krcmarık, Jan Koci, “Accurate object detection system on HoloLens using YOLO algorithm”, 2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO).

[7]. Samiksha Choyal, Ajay Kumar Singh,” An Acoustic based Roadside Symbols Detection and Identification using Faster RCNN and SSD”, 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3).

[8]. Narayana Darapaneni, Umang Maheshwari, Anwesh Reddy Paduri, Parikshit Bangade, Sushilkumar C Thorawade, Amit Mane, Rushikesh Borse, “Traffic Monitoring and Analysis At Toll Plaza”, 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS).

[9]. Min Zhou, Xiaodong Chen, Lei Li, Clive Parini, “The UWB Imaging System with Rotating Antenna Array for Concealed Metallic Object”, The 8th European Conference on Antennas and Propagation (EuCAP 2014).

[10]. Amit Ghosh, Shamsul Arefeen Al Mahmud, Thajid Ibna Rouf Uday, Dewan Md. Farid, “Assistive Technology for Visually Impaired using Tensor Flow Object Detection in Raspberry Pi and Coral USB Accelerator”, 2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020.

[11]. Weiwei Wang, Kehu Yang, “A METHOD FOR MILLIMETER-WAVE IMAGING OF CONCEALED OBJECTS VIA DE-ALIASING”, IEEE 2020.

[12]. Sagar Mahurkar, “Integrating YOLO Object Detection with Augmented Reality for iOS Apps”, IEEE 2018.

[13]. Gunasekaran Muthumanickam, Gopalakrishnan Balasubramanian, “A Traffic Congestion Control in Urban Areas with Vehicle-Infrastructure Communications”, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017).

[14]. Asad Ullah, Hongmei Xie2, Muhammad Omer Farooq, Zhaoyun Sun, “Pedestrian Detection in Infrared Images Using Fast RCNN”, 2018 IEEE.

[15]. Qimei Wang, Feng Qi, “Tomato diseases recognition based on Faster RCNN”, 2019 10th International Conference on Information Technology in Medicine and Education (ITME).

[16].kan Xu, Zheiwen Yuan, Jinli Zhan, Yiping Ji, Xing He, Haosen Yang, “SF6 Gas Infrared Thermal Imaging Leakage Detection Based On Faster RCNN”, 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA).

[17]. Feng Yang, Hang Yu, Kamolrat Silamut, Richard J Maude, Stefan Jaeger, Sameer Antani,” Parasite Detection in Thick Blood Smears Based on Customized Faster-RCNN on Smartphones”,2019 IEEE.

[18]. Yap Hong Yeu, Mohd Ibrahim Shapiai, Zool Hilmi Ismail, Hilman Fauzi, “Investigation on Different Colour Spaces on Faster RCNN for Night-Time Human Occupancy Modelling”, 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC 2019).

[19]. Harit Ahuja, Vedant Kuhar, R. I. Minu,” Object Detection and Classification for Autonomous Drones”, International Journal of Recent Technology and Engineering (IJRTE).

[20]. Wang Cheng, Peng Zhihao, “Design and Implementation of an Object Detection System Using Faster R-CNN”, 2019 International Conference on Robots & Intelligent System (ICRIS).

[21]. Suraj K Mankani, Naman S Kumar, Prasad R Dongrekar, Shreekant Sajjanar, Mohana, H V Ravish Aradhya,” Real-Time Implementation of Object Detection and Tracking on DSP for Video Surveillance Applications”, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016.