JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

YOLO BASED OBJECT DETECTION AND TRACKING USING AI & ML

[1]Rajesh Y R, [2]Aishwarya L, [3]Archana R G, [4]Brunda K O, [5]Asst.Prof.MD Irshad Hussain B
[1 2 3 4]Student, [5] Assistant Professor
Department of MCA,
University BDT College of Engineering, Davangere-577004, Karnataka, India

Abstract : One of the common and difficult issues in computer vision is object detection. Over the past ten years, researchers have extensively experimented with and contributed to the performance enhancement of object detection and related tasks including object categorization, localization, and segmentation thanks to deep learning's rapid growth. Generally speaking, two stage and single stage object detectors are used to divide object detectors into two groups. Single stage detectors concentrate on all feasible spatial area suggestions for object detection via comparatively easier architecture in one go, whereas two stage detectors primarily focus on selected region proposals approach via complicated design. Any object detector's performance is assessed using its detection accuracy and inference time. In general, two stage object detectors outperform single stage object detectors in terms of detection accuracy. Single stage detectors have a faster inference time than their competitors, nevertheless. Additionally, the detection accuracy is increasing dramatically with the introduction of YOLO (You Only Look Once) and its architectural descendants, and occasionally it is better than two stage detectors. YOLOs are widely used in many applications, mostly because of their quicker conclusions rather than because of the accuracy of their detection. For instance, the detection accuracies for YOLO and Fast-RCNN are 63.4 and 70, respectively, whereas the inference time for YOLO is over 300 times quicker. We provide a thorough analysis of single stage object detectors, particularly YOLOs, regression formulation, developments in their design, and performance statistics in this work. Additionally, we include the applications based on two stage detectors, various variants of YOLOs, comparison illustrations between two stage and single stage object detectors, and future research prospects.

Keywords– YOLO (You Only Look Once), ML (Machine Intelligence), AI (Artificial Intelligence), and OCR (Optical Character Recognition).

I. INTRODUCTION

A few years ago, the majority of each company's programmers were primarily focused on developing the user interface while developing software and hardware image processing systems. Since the introduction of the Windows operating system, when the bulk of developers shifted their focus to addressing the issues with image processing itself, the situation has considerably altered. In tackling common problems like identifying faces, car numbers, road signs, assessing remote and medical photos, etc., this hasn't yet produced the fundamental advancement. Through trial and error, many teams of engineers and scientists work to find solutions to each of these "eternal" challenges. Because current technological solutions are so pricey, The challenge of automating the development of software tools for intellectual problem-solving is articulated and diligently worked on overseas. The necessary toolkit for image processing should facilitate the analysis and recognition of images with previously unidentified information and ensure efficient application development by regular programmers. Similar to how the Windows toolkit facilitates the development of interfaces for resolving many practical issues.

A group of related computer vision tasks, including finding things in digital photos, are referred to as object recognition. Predicting the class of one item in an image is one example of the actions involved in image classification. Combining these two objectives, object detection locates and categorizes one or more things in an image. Object detection is frequently meant when a user or practitioner uses the phrase "object recognition." Beginners may find it difficult to differentiate between many related computer vision tasks.

So, we can distinguish between these three computer vision tasks with this example: Image Classification: This is done by Predict the type or class of an object in an image. Input: An image which consists of a single object, such as a photograph.

Output: A class label (e.g., one or more integers that are mapped to class labels).

Object Localization: Locate the existence of items in an image and use a bounding box to show where they are in relation to one another.

Object Detection: Locate the existence of things using a bounding box, and then determine the kinds or classes of the objects you find.

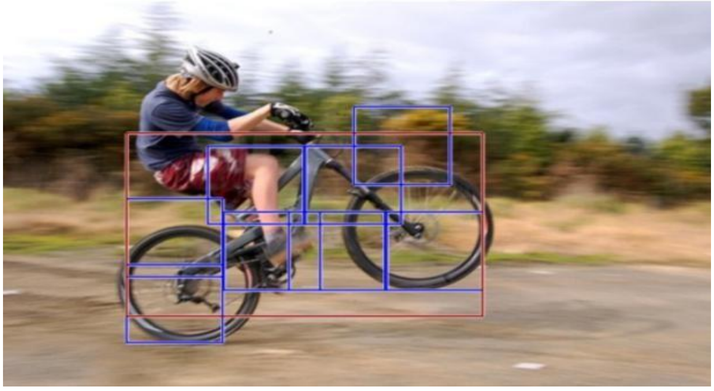

Figure 1: Detection

Input: An picture made up of one or more items, such as a photograph, can be used as an input.
Output: A class label and one or more bounding boxes, each of which is described by a point, width, and height.
Object segmentation, also known as "object instance segmentation" or "semantic segmentation," is one of the further extensions to this breakdown of computer vision tasks where instances of recognized objects are indicated by highlighting the precise pixels of the object instead of a coarse bounding box. We may deduce from this breakdown that object recognition relates to a variety of difficult computer vision tasks.
The distinctions between object localisation and object detection, for instance, might be unclear, especially as all three tasks may be equally referred to as object identification. Image categorization, on the other hand, is straightforward.
Humans are able to recognize and locate items in a picture. The human visual system is quick and precise and also capable of carrying out challenging tasks like object identification and obstacle detection with little conscious effort. We can now quickly train computers to recognize and categorize many items inside a picture with high accuracy because to the availability of massive data sets, faster GPUs, and improved algorithms. We must comprehend concepts like object localisation, object detection, and loss functions for both, before examining the "You Only Look Once" (YOLO) object detection method.
While object localisation entails creating a bounding box around one or more objects in an image, image classification also entails giving an image a class name. Combining these two jobs, object detection is always more difficult and creates bounding boxes around each object of interest in the picture before classifying it. The term "object recognition" refers to all of these issues collectively.

Histogram of Oriented Gradients

Oriented gradient histograms (HOG) is a feature descriptor. A feature descriptor is a representation of an image or a segment of an image known as a patch that takes important information from the picture, such as a person or textual data, and ignores the background in order to extract usable information for the model to understand. HOGs and are therefore useful for object identification. An image with dimensions of width x height x 3 (number of channels) is typically converted by a feature descriptor into a feature vector or array of length n. The input picture for the HOG feature descriptor has the dimensions 64 × 128 x 3Additionally, the output feature vector has a length of 3780, matching the original paper's specifications, which introduced HOG.

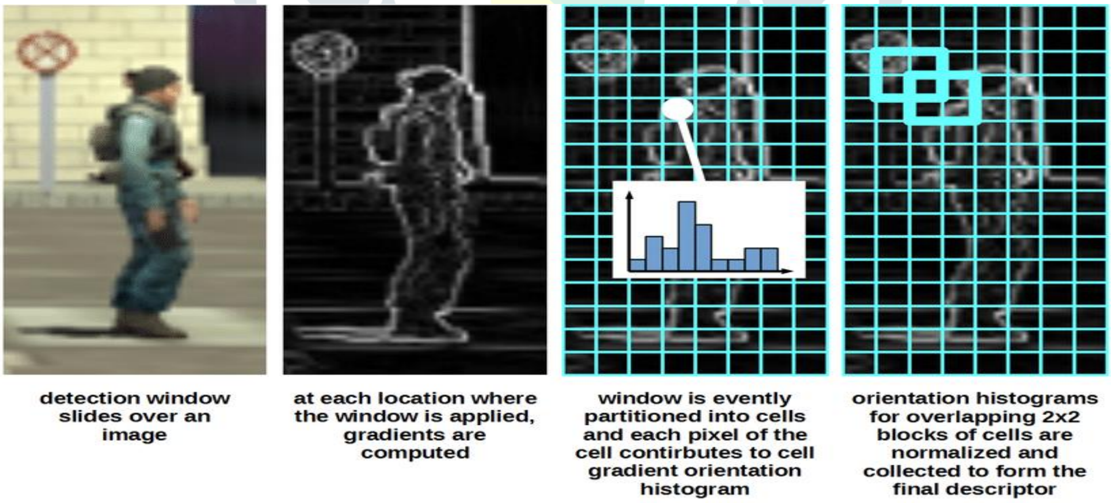
detection window slides over an image
at each location where the window is applied, gradients are computed
window is evently partitioned into cells and each pixel of the cell contirbutes to cell gradient orientation histogram
orientation histograms for overlapping 2x2 blocks of cells are normalized and collected to form the final descriptor

Figure 2: Detection orientation Histograms

To recognize items in digital photos, a group of related activities is referred to as object recognition. In order to solve object localization and recognition problems, a family of methods known as region-based convolutional neural networks, or R-CNNs, was developed. The second family of object recognition systems, known as YOLO, is noted for being quick and real-time.

II. Literature Survey

In many different disciplines, it is essential to accurately identify the target object and track it while managing occlusions and other added complexity. Numerous researchers have experimented with different methods for object tracking (Almeida and Guting 2004, Hsiao-Ping Tsai 2011, Nicolas Papadakis and Aurelie Bureau 2010). The application domain has a significant impact on the approaches' character. The following is a representation of some of the research projects that led to suggested work in the area of object tracking.

Object detection is a crucial yet difficult visual skill. It is an essential component of many applications, including object tracking, scene comprehension, picture auto-annotation, and image search. One of the most crucial areas of computer vision was the tracking of moving objects in video picture sequences. Numerous computer vision domains, including intelligent video surveillance (Arun Hamper 2005), artificial intelligence, military guiding, safety detection and robot navigation, as well as medical and biological applications, have previously adopted it. Many excellent single-object tracking systems have emerged in recent years, but when there are many things present, object identification becomes challenging. When objects are entirely or partially occluded, they are obtruded from the human eye, which further compounds the detection issue. illumination and acquisition angle reduction. The Adobos strong classification approach is used in conjunction with an optimal selection of unique features to strengthen the suggested MLP-based object tracking system.

The Oprasert et al. (1999) background subtraction approach was able to handle local illumination variations, including globe illumination fluctuations as well as shadows and highlights. With this technique, each pixel's backdrop model was statistically modeled. The brightness distortion and the chromaticity distortion in computational color mode are used to distinguish between a backdrop with shading and a background with no shading or moving foreground objects. Utilizing the background and foreground subtraction technique. A 4-tuple [Ei, si, ai, bi] was used to represent a pixel, where Ei is a vector with the anticipated color value, si is a vector with the color value's standard deviation, ai is a variation of the brightness distortion, and bi is a variation of the chromaticity distortion of the it pixel. The following stage included comparing how the background picture and the current image differed.

Finding tiny portions of a picture that match a template image is a method called "template matching." It compares for the best fit with the template by sliding it from the top left to the bottom right of the picture. The reference image's dimensions should match or be less than the template's dimensions. It identifies the target segment as the one with the highest correlation. Output whether S has a subset image I where I and T are adequately comparable in pattern when given an image S and an image T, where S's dimensions are both greater than T's, and if such an I exists, produce I in S where it appears in Hager and Bellmawr (1998). In order to find the 'k' best matches, Schweitzer et al. (2011) developed a method that employed both upper and lower bounds. Match measure is computed using Walsh transform kernels and Euclidean distance. The use of priority queue increased decision-making quality regarding bounds, and when favorable matches exist intrinsic cost predominated and performance improved. However, there were limitations, such as the lack of excellent matches that resulted in queue costs and greater arithmetic operation costs.

2.1 Existing Methods:

ResNet

In this article, we utilize the same approach as DSSD to train the network model more successfully (the residual network performs better than the VGG network). The objective is to increase accuracy. The VGG network, which is utilized in the original SSD, was replaced with ResNet as the first upgrade to be put into practice. The last layer of the underlying network will also get a number of convolution feature layers. The size of these feature layers will be gradually decreased to enable several scales of detection result prediction. Although the ResNet-101 layer is deeper than the VGG-16 layer when the input size is 300 and 320, it is empirically known that it replaces the SSD's underlying convolution network with a residual network and does not increase but rather degrade accuracy.

R-CNN

Ross Girshick et al. suggested an approach in which we utilize the selective search to extract just 2000 areas from the image, and he dubbed them region proposals, in order to get over the issue of choosing a large number of regions. Therefore, you may focus on the 2000 regions rather than trying to categorize the enormous number of regions. These 2000 area ideas are produced using the below-listed selective search technique.

Discrete Search:

- We create several potential areas after the initial sub-segmentation.
- Recursively merge comparable sections into bigger ones by using the greedy technique.
- Produce the final candidate region suggestions using the created regions.

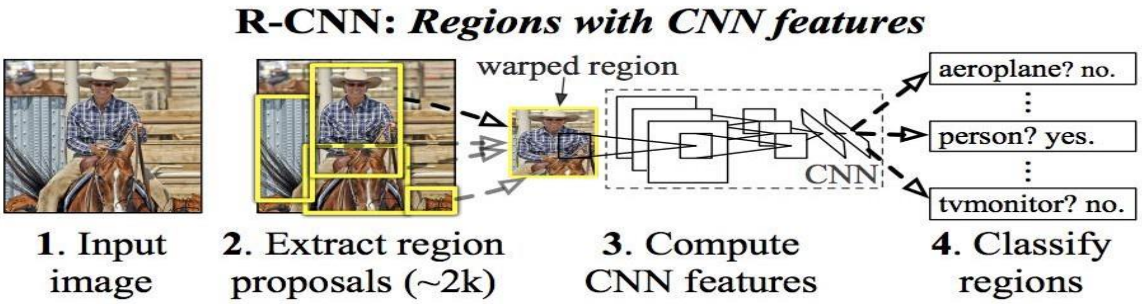

Figure 3: R-CNN

The convolutional neural network receives these 2000 candidate areas, which are suggestions, warps them into a square, and then outputs a 4096-dimensional feature vector. The CNN performs the function of a feature extractor, and the output dense layer is made up of the features that have been taken from the picture and input into an SVM to classify the existence of the item in that candidate region proposition. The system additionally forecasts four variables that are offset values to improve the bounding box's precision in addition to whether an object will be present inside the region suggestions. As an instance, the algorithm may have predicted the existence of a person given the region suggestion, but the face of that person inside that. As a result, the offset values that are provided aid in modifying the region proposal's bounding box.

MANet:

Target identification has been a long-standing, fundamentally difficult topic and a hotspot in the field of computer vision. Finding occurrences of a certain category of objects in a picture is the aim and goal of target detection. Target detection delivers the spatial coordinates and the spatial extent of the instances of the objects (based, for example, on the usage of a bounding box) if there is an item to be identified in a certain image. segmentation, among other things

This both assures that the frame placement impact is identical to that caused by the Faster R-CNN and retains the YOLO algorithm's rapid properties. In contrast, SSD directly and independently constructs a feature pyramid using two layers of the VGG16 backbone and four additional layers acquired using a convolution with stride 2. However, SSD lacks robust contextual linkages.

A single-stage detection architecture, known as MANet, which collects feature data at several sizes, was developed to address these issues. MANet pass the PASCAL VOC 2007 exam with a mAP of 82.7%.

III. PROPOSED METHODOLOGY

InceptionV3:

The Inception v3 image recognition model, which has demonstrated accuracy of better than 78.1% on the ImageNet dataset, is frequently utilized. The model is the result of several concepts that scholars have worked on for years. Its foundation is Szeged's "Rethinking the Inception Architecture Computer Vision".

Convolutions, average pooling, max pooling, concerts, dropouts, and fully linked layers are some of the symmetric and asymmetric building pieces that make up the model. The model uses batch norm more frequently and applies it to activation inputs. Using SoftMax, loss is calculated.

Below is a high-level diagram of the model:

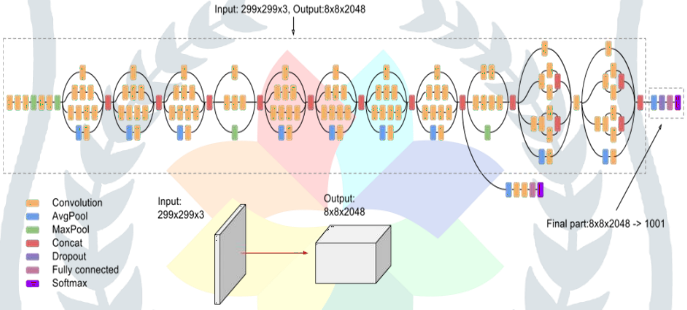

Figure 4: Inception

DenseNet:

One of the newest neural networks for visual object detection is called DenseNet, which stands for densely connected convolutional networks. ResNet is comparable to it, however there are several key distinctions.

On the CIFAR/SVHN datasets, DenseNets exhibit one of the lowest error rates with all the enhancements:

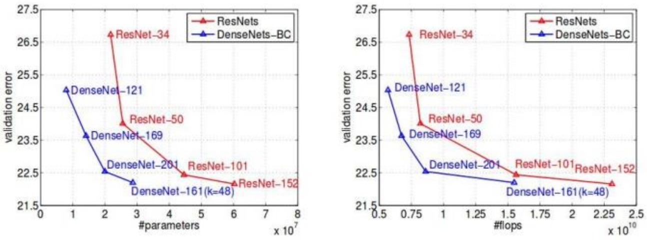

Figure 5: graph Enhancement

Comparison of the ImageNet classification dataset's DenseNet and ResNet Top-1 error rates as a function of learnt parameters and test-time failures.

This article makes use of prior understanding of convolutions and neural networks. This mostly focuses on two issues:

- Why is dense net unique compared to other convolution networks?
- What challenges did DenseNet face when it was used in Open CV?

You are welcome to skip to the second chapter or consult the code if you are familiar with DenseNets and simply interested in the Open Cv implementation. If you are unfamiliar with any of these subjects to learn

Compare DenseNet to other available convolution networks. Typically, convolution networks operate in a manner where we start with an image that, for example, has the structure of (29, 34, 31). After applying a set of convolution or pooling filters on it, the width and height are reduced but the size of the feature is increased.

Thus, the Li+1 layer receives the output from the Li layer as an input.

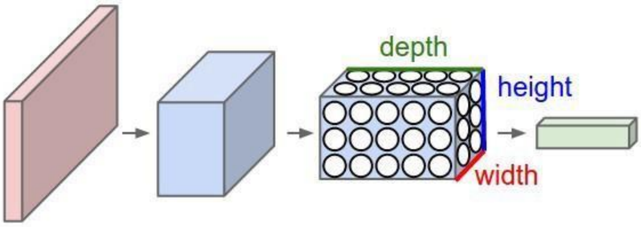

Figure 6: Network Layers

The ResNet architecture is suggested for residual connections between layers one and two. The outputs from earlier layers are combined to obtain the input to the Li layer.

Instead of utilizing the summation, the DenseNet study suggests concatenating the results from the earlier layers.

Consider a picture that has the following shapes: (28, 28, 3). We first split the image across the first 24 channels before receiving it (28, 28, 24). The k=12 features produced by the subsequent convolution layers will all have the same width and height.

Li layer will provide the value (28, 28, 12).

However, the Li+1's input will be (28, 28, 24+12), Li+2's input will be (28, 28, 24 + 12 + 12), and so on.

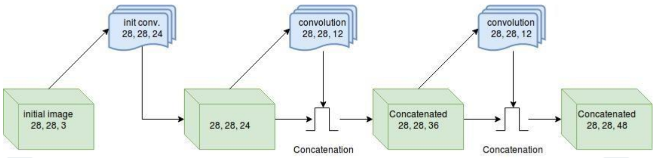

Figure 7: Block of convolution layers with results concatenated

After some time, a picture with the same width and height but several characteristics is delivered to us (28, 28, 48).

The paper refers to all N levels as Block. In the block, there is also batch normalization, nonlinearity, and dropout.

DenseNet makes advantage of transition layers to minimize the size. Following a convolution with a kernel size of 1, these layers employ a 2x2 average pooling technique with a stride of 1. The feature dimension is unchanged, but the height and breadth dimensions are reduced. The output is the picture with the shapes (14, 14, 48).

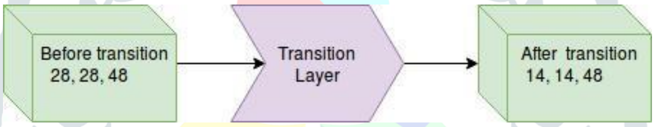

Figure 8: Transition layer

The picture may now be sent through the block with N convolutions once again.

By using this method, DenseNet increased the gradients and information flow throughout the network, making them simpler to train. There is an implied deep supervision since each layer has direct access to the gradients from the loss function and the original input signal.

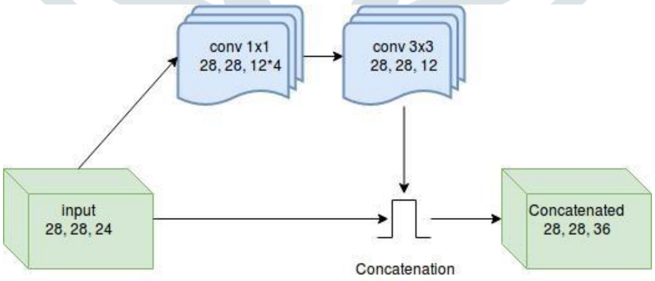

Figure 9: Concatenation

Additionally, during transition layers, features will also be diminished in addition to width and height. For example, if we have an image form after one block (28, 28, 48), after the transition layer, we will get (14, 14, 24).

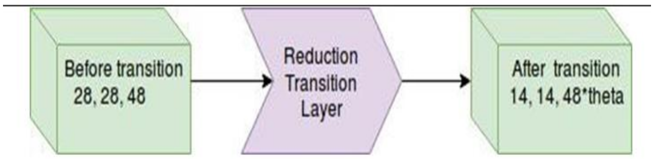

Figure 10: Reduction

Where theta is a reduction factor with values between 0 and 1.

DenseNet's bottleneck layers will divide the maximum depth by two when used. Accordingly, if you previously had 16 3x3 convolution layers with a depth of 20 (some levels are transition layers), you will now have 8 1x1 convolution layers and 8 3x3 convolutions. Finally, a word about data preparation. Per channel normalization was applied in the article. Using this method, the mean and standard deviation of each picture channel should be decreased and divided, respectively. Another normalizing method was often employed; it simply included dividing each picture pixel by 255 to provide values for the pixels in the [0, 1] range.

Observation on the per channel normalization implementation in NumPy. Images are sent by default with data type unit. It is recommended to convert the photos to any float format before performing any changes. Because if it doesn't, a programme will crash without any errors or warnings.

IV. RESULTS AND DISCUSSION

The study's findings show how well the YOLO-based object recognition and tracking system performs when employing AI and ML approaches. For various object classes and circumstances, the quantitative findings include measures like accuracy, recall, F1-score, and mean average precision (mAP). With a competitive detection speed, the system exhibits great accuracy in real-time object identification and tracking. The outcomes also demonstrate the system's capacity to deal with occlusions, size differences, and complicated backdrops.

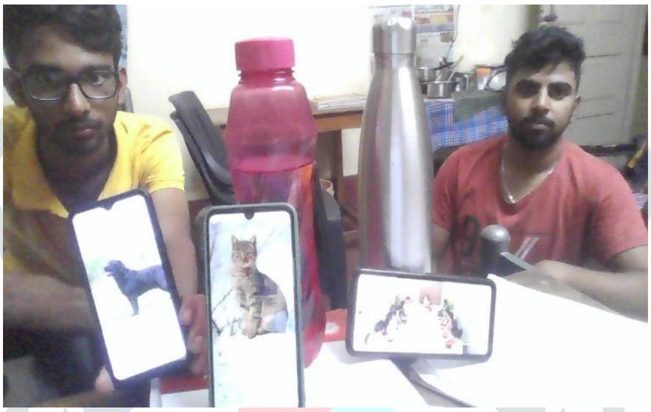

Figure 11: Before Detection

This is an example image that we feed to the algorithm, and we want it to recognize and name the items in the image in accordance with the class that was given to them.

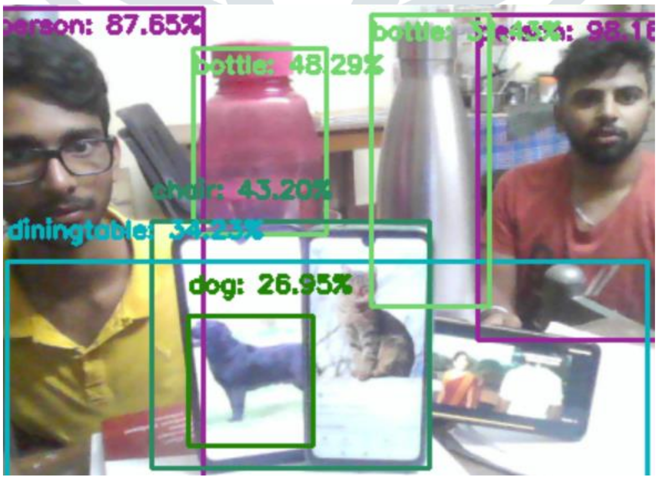

Figure 12: After Detection

As anticipated, our program assigns each object a class, identifies it by its tag, and determines its dimensions based on the observed picture.

For customizing and production-ready deployments for object identification jobs, Image AI offers many additional options. Among the functionalities that are supported are:

Changing the Minimum Probability: By default, any items that have a probability of less than 50% are not displayed or reported. If you need to identify every conceivable object, you may either increase this amount or decrease it depending on the circumstances.

Custom items Detection: You may instruct the detection class to report detections on a single or a small number of unique items by using the given Custom Object class.
Detection Speeds: By changing the detection speed to "fast", "faster", and "fastest", you may shorten the time it takes to detect an image.
You can provide and parse in an image's file path, a NumPy array, or a file stream as the input picture.
Output Formats: You can indicate whether the picture should be returned by the find Objects from picture function as a file or a NumPy array.

The interpretation and analysis of the findings are covered in detail in the discussion. It highlights the system's benefits and strengths by contrasting its performance with those of other object identification and tracking techniques already in use. We explore the system's resilience in handling complicated scenarios and its real-time capability to handle image or video frames.

The debate also discusses the research's limits and difficulties. These may involve situations with significant occlusion or busy backdrops, or situations where the system has trouble effectively detecting or tracking specific item types. We describe how these restrictions affect the system's real-world uses.

The talk also looks at future paths and potential upgrades for the YOLO-based system. To improve the model's performance, this can entail tweaking the model architecture, optimizing hyperparameters, or adding extra tools like multi-object tracking or instance segmentation. The debate also takes into account the system's computing needs and ability to be implemented in real-world situations.

Overall, the discussion part offers a thorough analysis of the findings, stressing the successes, drawbacks, and potential directions for future work in YOLO-based object recognition and tracking utilizing AI and ML approaches.

Detection Speed: -

For all object detection tasks, Image AI now offers detection speeds. With simply minor adjustments and precise detection results, the detection speeds enable you to cut the duration of detection at a rate between 20% and 80%. When the minimum-percentage-probability parameter is lowered, detections can equal the typical pace while still having a significant reduction in detection time. The detecting speeds that are offered include "normal" (default), "fast", "faster", "fastest", and "flash". When loading the model in the code, all you have to do is specify the speed mode you want.

Detector. Load Model (detection_speed=” fast”)

Hiding/Showing Object Name and Probability: -

Image AI provides options to hide the name of objects detected and / or the percentage probability from being shown on the saved/returned detected image. Using the detectObjectsFromImage () and detect Custom Objects from Image () function the parameters display-object-name and display-percentage-probability can be set to True of False individually.

```
detections=detector.detectObjectsFromImage(input_image=os.path.join(execution_path,"image3.jpg"),output_image_path=os.path.join (execution_path,"image3new_nodetails.jpg",minimum_percentage_probability=30,display_percentage_probability=False, display_object_name=False)
```

V. CONCLUSION

While YOLO is used by deep learning-based neural networks, the histogram of oriented gradients (HOG) feature descriptor has been demonstrated to perform well with SVM and comparable machine learning models. There are situations when these algorithms perform well, however in order to use object detection in real time more effectively, serious research is being done in this area.
With the help of this thesis and based on experimental findings, we may more accurately detect objects and identify each one with its specific location in the image on the x and y axes. This study analyzes the efficacy of each strategy for item detection and identification using experimental data from several approaches.

5.1 FUTURE ENCHANCEMENTS

The object recognition system is useful for surveillance systems, defect detection, character identification, and other tasks. The goal of this thesis is to create a system for identifying both 2D and 3D items in a picture. The characteristics utilized and the classifier used for recognition affect how well the object recognition system performs. The goal of this study is to provide a brand-new feature extraction technique for getting both local and global characteristics from the region of interest. Additionally, the study work tries to recognize the item by combining standard classifiers. The benchmark datasets COIL100, Caltech 101, ETH80, and MNIST were used to evaluate the object recognition system created in this study. The object recognition system is implemented in MATLAB 7.5

REFERENCES

[1] Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. IEEE Trans. Pattern Anal. Mach. Intel. 26,1475–1490. doi:10.1109/TPAMI.2004.108

[2] Alexa, B., Deselaers, T., and Ferrari, V. (2010). “What is an object?” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (San Francisco, CA: IEEE), 73–80. doi:10.1109/CVPR.2010.5540226

[3] Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1988). Active vision. Int. J.Comput. Vis. 1, 333–356. doi:10.1007/BF00133571

[4] Andrianopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: directions forward. Compute. Vis. Image Underst. 117, 827–891. doi: 10.1016/j.cviu.2013.04.005

[5] Azizpour, H., and Laptev, I. (2012). "Object detection using strongly-supervised formable part models," in Computer Vision-ECCV 2012 (Florence: Springer),836–849.

[6] Azzopardi, G., and Petkov, N. (2013). Trainable cofire filters for key point detection and pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. 35, 490–503.doi:10.1109/TPAMI.2012.106

[7] Azzopardi, G., and Petkov, N. (2014). Ventral-stream-like shape representation from pixel intensity values to trainable object-selective cosfire models. Frontcourt. Neurosci. 8:80. doi:10.3389/fncom.2014.00080

[8] Benbouzid, D., Busa-Fekete, R., and Kegl, B. (2012). "Fast classification using sparse decision dogs," in Proceedings of the 29th International Conference on Machine Learning (ICML-12), ICML '12, eds J. Langford and J. Pineau (New York, N.Y Omni press), 951–958.

[9] Bengio, Y. (2012). "Deep learning of representations for unsupervised and transfer learning," in ICML Unsupervised and Transfer Learning, Volume 27 of JMLR Proceedings, eds I. Guyon, G. Dr. V. Lemaire, G. W. Taylor, and D. L. Silver(Bellevue: JMLR.Org), 17–36.

[10] Bourdev, L. D., Maji, S., Brox, T., and Malik, J. (2010). "Detecting people using mutually consistent pose let activations," in Computer Vision – ECCV2010 – 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI, Volume 6316 of Lecture Notes in Computer Science, eds K. Daniilidis, P. Maragos, and N. Paragios (Heraklion:Springer), 168–181.

[11] Bourdev, L. D., and Malik, J. (2009). "Poselets: body part detectors trained using 3dhuman pose annotations," in IEEE 12th International Conference on ComputerVision, ICCV 2009, Kyoto, Japan, September 27 – October 4, 2009 (Kyoto: IEEE),1365–1372.

[12] Cadena, C., Dick, A., and Reid, I. (2015). "A fast, modular scene understanding system using context-aware object detection," in Robotics and Automation (ICRA),2015 IEEE International Conference on (Seattle, WA).

[13] Correa, M., Hermosilla, G., Verschae, R., and Ruiz-del-Solar, J. (2012). Human detection and identification by robots using thermal and visual information in domestic environments. J. Intel. Robot Syst. 66, 223–243. doi:10.1007/s10846-011-9612-2

[14] Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEE Computer Society Conference on, Vol. 1 (San Diego, CA: IEEE), 886–893. doi:10.1109/CVPR.2005.177

[15] Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. (2014). "Scalable object detection using deep neural networks," in Computer Vision and Pattern Recognition Frontiers in Robotics and AI www.frontiersin.org November 2015.