International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 25 Issue 01, JAN, 2025

OBJECT DETECTION AND RECOGNITIONS USING WEBCAMS WITH VOICE USING YOLO ALGORITHM

Ms.P.Sudeshna[1],Garimella. Krishna Karthika[2],Kethavath. Prathyusha[3],Korivi Indhu[4]

[1] Associate Professor, School of CSE ,Malla Reddy Engineering College For Women(Autonomous Institution), Maisammaguda, Dhulapally,Secunderabad,Telangana-500100
[234]UG Student, Department of CSE,Malla Reddy Engineering College for Women, (Autonomous Institution), Maisammaguda,Dhulapally,Secunderabad,Telangana-500100

ABSTRACT

Object detection from deep learning has performed better in many applications, and yet real images usually depict challenges such as noise blurring or jitter rotation. Such problems generally degrade the accuracy and the efficiency of object detection systems dramatically. The purpose of this work is to identify objects satisfactorily by applying a method called You Only Look Once (YOLO), which tackles numerous problems while it provides plenty of advantages over other solutions.Unlike other algorithms that are known as Convolutional Neural Networks (CNN)or Fast-Convolutional Neural Networks, process images partially by focusing on specific areas. YOLO is different from that; it looks at an entire image at once while predicting bounding boxes and class probabilities for objects within these boxes by making a single forward pass of the network. This holistic approach makes YOLO significantly faster and much more efficient than other algorithms that process parts of the image separately.For this project, we have used the YOLO algorithm for detecting all types of objects in a real-time scenario. Moreover, an Android Application has been developed incorporating YOLO-based object detection for enhancing user experience. This application does not only detect objects in images or live video feeds but also provides voice feedback to the user. Such an enhancement would make the system much more accessible for individuals with visual impairments because they could receive immediate auditory information about the objects detected in their environment.

Overall, YOLO's speed, accuracy, and versatility make it a great choice for real-time object detection applications, and the integration with a mobile application provides a practical solution for various use cases.

KEYWORDS: Object Detection,Deep Learning,YOLO ,Real-time Detection,Bounding Boxes.

I INTRODUCTION

Object detection is a basic computer vision technique that identifies and locates objects in an image or video. This is different from image recognition, which labels an entire image with a class and does nothing further than identifying which object is actually present but says nothing about the location. It accomplishes this by drawing a bounding box around each object it has recognized, providing a clear and accurate sense of where exactly these objects exist. In short, the above difference in the process implies how, for example, while image recognition would only name the given picture as an "apple, an object recognition will sketch a bounding box around multiple apples found in a given picture. The applications of object detection are widespread and go beyond the scope of industries such as security surveillance, autonomous driving, health, retail, and robotics. It plays a core role in automating tasks which involve identifying and tracking objects in dynamic environments. However, object detection has challenges. Different environmental conditions such as fog, rain, and low light can significantly affect the performance of detection algorithms. Moreover, processing time and detection accuracy are vital aspects that need to be taken care of in real-world applications. Some algorithms have been developed that can overcome these challenges in improving the efficiency and accuracy of object detection. These include Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, Histogram of Oriented Gradients (HOG), Region-based Fully Convolutional Networks (RFCN), Single Shot Detector (SSD), Spatial Pyramid Pooling (SPP-net) and YOLO (You Only Look Once). Each of these algorithms has its advantages and disadvantages in terms of speed and accuracy. For instance, techniques that rely on R-CNN are usually highly accurate but tend to be very computer intensive. On the other hand, YOLO is known for its speed but may sacrifice some accuracy. To address the challenges of accuracy and computational performance, various researchers have proposed new architectures and improvements. In this paper, we propose a new approach to improve object detection in video streams by combining the strengths of existing algorithms while minimizing their shortcomings. This approach tries to provide more reliable real-time object detection capabilities for various applications by improving the detection process, reducing computation time, and taking into account environmental challenges

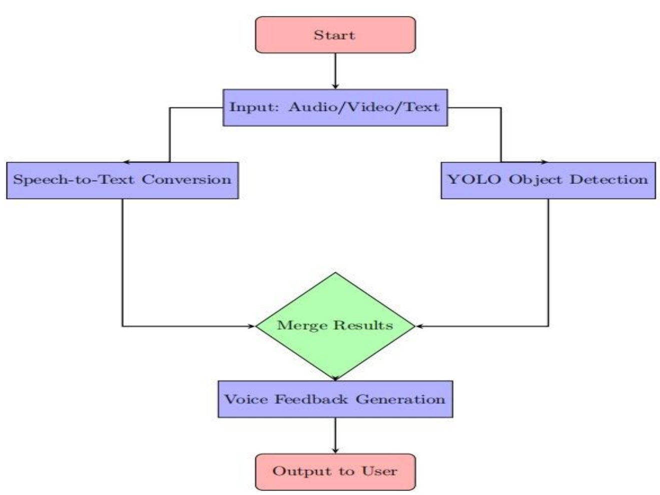

Fig 1:System architecture

II.RELATED WORK

1. Nguyen et al. give a general framework for robust detection, recognition, and tracking of pedestrians and vehicles on roads. They merge the strengths of deep learning with the strengths of using multiple local patterns and depth information, which can be exploited efficiently. The detected obstacles in t he next frame were tracked by a tracking-based deep learning algorithm. It was then implemented using a GPU for real-time use. Through various experiments to determine the performance, this framework provides good detection, recognition and tracking for real-time driving assistance. It can then be extended to detect and recognize other objects such as traffic signs and traffic lights. This article was about obstacle detection, recognition and tracking. The object behavior can be predicted for future research.

2.Deng H . Sun et al. proposed a region-based CNN approach for the vehicle detection in aerial images. They integrated two CNNs: AVPN for hierarchical feature maps fusion that is more accurate for small objects detection and VALN for attribute annotation.

They experimented on various datasets and obtained good results on images from UAV or Google Earth. However, there were still some false positives. Accurate vehicle detection remains a challenging challenge. Future work will focus on hard mining We use a bootstrap strategy and multi-GPU configuration to handle negative samples and reduce computation time.

3.Y. Cai et al. use a scene-adaptive vehicle detection algorithm and a bagging mechanism called bootstrap aggregation in order to generate numerous classifiers and to produce confidence scores for target training samples through voting. Then, using the automatic feature extraction function of DCNN and similarity calculation of source and target scene features with deep autoencoder, a composite deep structure-based scene-adaptive classifier and its training method are developed. Experiments have shown that This approach has the merits of high automation and a high detection rate of vehicles. Limitations The method is that the reliability assignment method is a simple linear function that depends on the members of the subclassifier, which is relatively subjective and has no theoretical basis.

4.Cao et al. proposed novel framework for deep neural networks in the student network is trained to minimize the combination loss required for domain adaptation, and knowledge distillation is also designed at the same time. Besides a smaller and faster DCNN model, the detection side also has the following features: We developed a faster method for determining candidates for the region of interest that contains the target object. We developed an analytical model for computing the support region of each convolutional layer and employed this We integrated a pre-existing object detection framework with a deep convolutional neural network. Experimental results on vehicle detection The video showed that the proposed method can speed up the network by up to 16 times while improving the object detection performance maintained.

5.Sommer et al. used Fast R-CNN and Faster R-CNN for vehicle detection in aerial images and evaluated the impact of both detection frameworks. Region Proportional Network (RPN) achieved the best performance among all proposed methods. They proposed a unique network optimized to handle small objects. Selective search gives the best results based on handcrafted features. The best detection performance was achieved with Faster R-CNN, which shares convolutional layers with RPN.

III.IMPLEMENTATION

A critical step in setting up this YOLO algorithm, which is really known for its efficiency in real-time object detection, for use in an Android application, which will implement object detection using voice feedback with a view of ensuring that the system operates as efficiently as possible to achieve an end user experience of comfort and relief especially for the visually impaired user.

First in the process: Setting up the YOLO Algorithm. YOLO works as it scans the whole image with one pass rather than the classical object detection approach where an image has to be divided into its regions to scan. So, it becomes a favorite for the applications that necessitate the rapid object detection capability. The installation of development environment involves setting up required libraries including OpenCV, TensorFlow, Keras, and NumPy by the package manager `pip`. These libraries are used to load the pre-trained YOLO model and perform the object detection tasks.

After setting up the dependencies, the next step is to download the pre-trained YOLO weights (`yolov3.weights`),the configuration file (`yolov3.cfg`), and the class labels (`coco.names`). These files are important because they define the architecture of the YOLO model and provide the class labels for the objects that the model can detect. The model is loaded into the Python environment using OpenCV, and the image is preprocessed—resized and normalized—before being passed through the YOLO model for detection. The model predicts the bounding boxes and class labels of the objects detected in the image.

The second step is to integrate this YOLO object detection system into an Android application. Using Android Studio, a new project is created while providing the required dependencies for OpenCV or TensorFlow Lite. TensorFlow Lite is used for the mobile-based optimization because the model size reduces and performance increases on mobile. YOLO's model is converted into TensorFlow Lite format, which would help in doing the processing efficiently. The Android app captures live video frames based on the Camera2 API. It then passes every frame in front of the YOLO model to detect objects.

The final integration is to make the app use the TTS Android API to supply live audio feedback of what's found by the app. For an object identified, it uses its detected class label as speech feedback on the name, "Apple," or "Car." A TTS engine needs to be initialized in the application so when an object has been identified, the application instantaneously offers feedback, making the app even more accessible for a user who is not in their vision.

In summary, this implementation combines real-time object detection with accessibility features. With the use of the YOLO algorithm for object detection and integration of voice feedback through

TTS, the app provides an effective and user-friendly solution to identify objects in real time. This makes the technology more accessible for people with visual impairments, making the overall functionality and inclusiveness of the application even better.

IV .ALGORITHMS

Algorithm for YOLO-based Object Detection with Voice Feedback

Initialize the YOLO Model

At first, the YOLO model is initialized by loading the pre-trained YOLO weights (`yolov3.weights`), configuration file (`yolov3.cfg`), and the class labels (`coco.names`). The configuration and weights of YOLO are a must for the recognition of different objects. By using the OpenCV or TensorFlow Lite (for Android devices), these models are loaded into the memory by configuring the object detection application.

Image Preprocessing

After the image or video frame is captured, they need to be preprocessed to meet the YOLO model's input specs. In other words, each image needs to have its dimensions resized to 416x416 and be normalized pixel values, ranging between [0, 1]. The image is, then, transformed to a "blob". A blob is a form of input which the YOLO model can work with. The blob is then passed through the YOLO network.

Object Detection

The preprocessed image is passed through the YOLO network for object detection. The model performs a forward pass and returns bounding boxes, class labels, and confidence scores for each detected object. A confidence threshold filters low-confidence detections. If the object's confidence score is above this threshold, then it is assumed to be detected, where the bounding box and the label of the object are retrieved for subsequent operations.

Android Implementation of Real-Time Object Detection

For Android implementation, the Camera2 API is used to capture live video frames. Every frame is preprocessed, resized, and normalized to fit the input requirements of YOLO. The frame is passed through the YOLO model for object detection, and bounding boxes are drawn around the detected objects. These results are displayed in real-time on the Android screen.

Text-to-Speech (TTS) for Voice Feedback

The detected object labels are converted into speech using the Android Text-to-Speech (TTS) API. When an object is identified, the application speaks the name of the object aloud (e.g., “Apple”, “Car”). This provides immediate audio feedback, making the application more accessible, particularly for users with visual impairments.

Continuous Detection

The system continuously captures video frames from the camera and processes them in real time. The YOLO model detects objects on each frame and provides voice feedback for each of the detected objects. This is performed for every new frame captured; hence, the detection and availability for users continue uninterrupted.

This algorithm details how combining YOLO object detection with voice feedback will result in an algorithm that ensures performance to be real-time while staying accessible to visually impaired people.

V.RESULTS

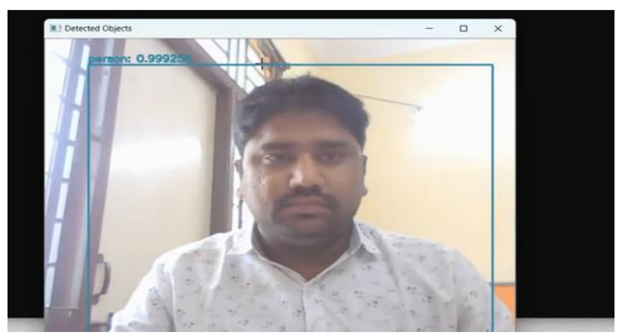

Fig 1:Detected as Person

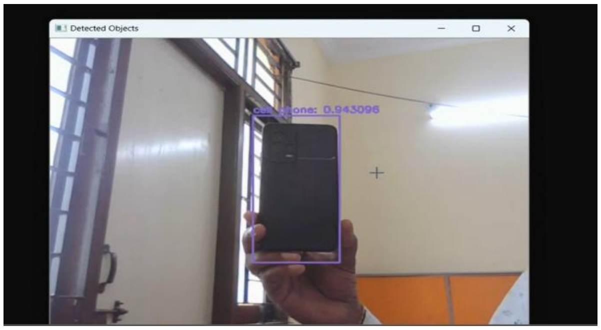

Fig 2:Detected Objected as Phone

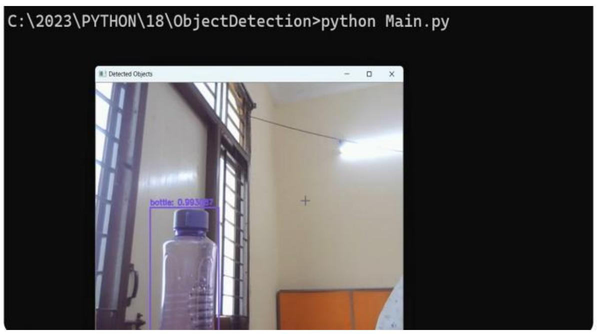

Fig 3:Detected Objected as Bottle

VI.CONCLUSION

In this project, the YOLO, with the Darknet framework and for real-time object detection, was successfully implemented in such a way that the YOLO model could process up to 67 frames per second on account of Darknet's high efficiency and flexibility and is thus best suitable for time-sensitive applications related to surveillance, automation, and accessibilities. Through extensive testing on a custom dataset captured in office and indoor environments, the model showed excellent detection accuracy for a wide range of objects, including desks, chairs, computers, and other office-related items.

This project shows that YOLO-Darknet is an efficient and scalable solution in real-time object detection under dynamic and adverse conditions. The results validate the potential of the approach in practical, real-world applications, especially in fields such as security surveillance, automation, and enhancing accessibility for visually impaired individuals. The successful integration of YOLO with Darknet presents a remarkable framework upon which various industries can realize their potential in using computer vision technology to build smarter, more efficient systems.

REFERENCES

[1] R. Rambola, P. Varshney and P. Vishwakarma, “Data Mining Techniques for Fraud Detection in Banking Sector,” 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-5, doi: 10.1109/CCAA.2018.8777535.

[2] N. Malini and M. Pushpa, “Analysis on credit card fraud identification techniques based on KNN and outlier detection,” 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 255-258, doi: 10.1109/AEEICB.2017.7972424.

[3] Ishan Sohony, Rameshwar Pratap, and Ullas Nambiar. 2018. Ensemble learning for credit card fraud detection. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD ’18). Association for Computing Machinery, New York, NY, USA, 289–294. DOI:https://doi.org/10.1145/3152494.3156815

[4] C. Wang, Y. Wang, Z. Ye, L. Yan, W. Cai, and S. Pan, “Credit Card Fraud Detection Based on Whale Algorithm Optimized BP Neural Network,” 2018 13th International Conference on Computer Science Education (ICCSE), Colombo, 2018, pp. 1-4, doi: 10.1109/ICCSE.2018.8468855

[5] I. Benchaji, S. Douzi and B. ElOuahidi, ”Using Genetic Algorithm to Improve Classification of Imbalanced Datasets for Credit Card Fraud Detection,” 2018 2nd Cyber Security in Networking Conference (CSNet), Paris, 2018, pp. 1-5, doi: 10.1109/CSNET.2018.8602972.

[6] John O. Awoyemi, Adebayo Olusola Adetunmbi, and Samuel Adebayo Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing

Networking and Informatics (ICCNI), pages 1–9, 2017.

[7] Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-A¨el Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi. Scarff: a scalable framework for streaming credit card fraud detection with spark. Information Fusion, 41:182–194, 2018.

[8] Galina Baader and Helmut Krcmar. Reducing false positives in fraud detection: Combining the red flag approach with process mining. International Journal of Accounting Information Systems, 2018.

[9] Ravisankar P, Ravi V, Raghava Rao G, and Bose, Detection of financial statement fraud and feature selection using data mining techniques, Elsevier, Decision Support Systems Volume 50, Issue 2, p491-500 (2011) SVM

[10] K. Seeja, and M. Zareapoor, "FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining," The Scientific World Journal, 2014, pp. 1-10. KNN, SVM

[11] C. Tyagi, P. Parwekar, P. Singh, and K. Natla, "Analysis of Credit Card Fraud Detection Techniques," Solid State Technology, vol. 63, no. 6, 2020, pp. 18057-18069. Credit card faud

[12] C. Chee, J. Jaafar, I. Aziz, M. Hassan, and W. Yeoh, "Algorithms for frequent itemset mining: a literature review," Artificial Intelligence Review, vol. 52, 2019, pp. 2603–2621. Litrature review AI

[13] S. Kiran, J. Guru, R. Kumar, N. Kumar, D. Katariya, and M. Sharma, "Credit card fraud detection using Naïve Bayes model based and KNN classifier," International Jounral of Advance Research, Ideas and Innovations in Technology, vol. 4, 2018, pp. 44-47. KNN Naïve Byers

[14] Pumsirirat, A.; Yan, L. Credit Card Fraud Detection Using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. Available online: https://thesai.org/Downloads/Volume9No1/Paper_3- Credit_Card_Fraud_Detection_Using_Deep_Learning.pdf (accessed on 23 February 2021). DL

[15] PwC's Global Economic Crime and Fraud Survey 2020. Available online: https://www.pwc.com/fraudsurvey (accessed on 30 November 2020). Fraud surver.

[16] Pourhabibi, T.; Ongb, K.L.; Kama, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. 2020, 133, 113303. Fraud detection.

[17] Lucas, Y.; Jurgovsky, J. Credit card fraud detection using machine learning: A survey. arXiv 2020, arXiv:2010.06479. Credit card fraud.

[18] Podgorelec, B.; Turkanovi´c, M.; Karakati˘c, S. A Machine LearningBased Method for Automated Blockchain Transaction Signing Including Personalized Anomaly Detection. Sensors 2020, 20, 147. Anomaly detection.

[19] Synthetic Financial Datasets for Fraud Detection. Available online: https://www.kaggle.com/ntnu-testimon/paysim1 (accessed on 30 November 2020). Fraud detection.

[20] Ma, T.; Qian, S.; Cao, J.; Xue, G.; Yu, J.; Zhu, Y.; Li, M. An Unsupervised Incremental Virtual Learning Method for Financial Fraud Detection. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–6. Financial fraud detection.