INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Image Detection and Real Time Object Detection

Rushikesh Lakhotiya[1], Mayuresh Chavan[2], Satwik Divate[3], Soham Pande[4]

[1, 2, 3, 4]*Student, Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, Maharashtra,India*

Abstract: Object detection is a crucial task in computer vision with various practical applications, including surveillance, autonomous vehicles, and robotics. The YOLO (You Only Look Once) algorithm is a popular real-time object detection algorithm that has gained significant attention due to its high accuracy and speed. This algorithm processes the entire image at once and predicts bounding boxes and class probabilities for identified objects, making it ideal for time-sensitive applications. YOLO has evolved through various versions, with YOLOv5 being the latest and most advanced version that employs a feature pyramid network (FPN) and anchor boxes to improve its object detection accuracy. In this project, we aim to implement YOLOv5 for real-time object detection and image detection tasks. We will train the model on a suitable dataset and evaluate its performance on various benchmarks, comparing it with other advanced object detection algorithms. The project's outcome will provide a robust and efficient solution for real-time object detection that can aid quick decision-making in identifying object categories and their respective positions. It has practical applications in surveillance, automated driving, and robotics.
Keywords: Object detection, computer vision, YOLO (You Only Look Once), OpenCV

I. INTRODUCTION

Computer vision involves two important tasks- identifying images and detecting objects in real-time. Image detection refers to the process of identifying within an image, while real-time object detection involves identifying and tracking objects in real-time video streams. Both of these tasks are essential in various fields, including autonomous vehicles, robotics, surveillance systems, and many more. The YOLO algorithm is a well-known approach utilized for detecting objects. The YOLO algorithm is a system for detecting objects in real-time, which utilizes only one neural network to anticipate the probability of object classes and bounding boxes in images or videos.In this paper, we will provide an introduction to YOLO and discuss its use in image detection and real-time object detection.

Traditional object detection algorithms rely on the use of sliding windows to search for objects in an image. However, this method can consume a lot of computational resources, making it computationally demanding, especially when working with large images or video streams. YOLO, in contrast, employs a solitary neural network to generate predictions for all the dataobjects in an image simultaneously, making it significantly faster than traditional methods. The YOLO algorithm works by dividing an image into a grid of cells and predicting both the probability of object classes and bounding boxes for every cell.

Recent advances in deep learning have led to significant improvements in image detection and real-time object detection algorithms. Convolutional neural networks (CNNs) are generally utilized in these algorithms to extract features from images or frames in videos, and then use these features to identify and locate objects in the scene.

Real-time object detection algorithms are designed to work in real-time, with the goal of detecting objects in a video stream as quickly and accurately as possible.

Overall, Real-time object detection and image detection are essential elements of various computer vision systems with numerous practical uses. Continued research in this field is expected to result in further advancements in accuracy, speed, and efficiency, leading to more advanced computer vision systems that can be applied in diverse scenarios.

II. LITERATURE REVIEW

The "Microsoft COCO: Common Objects in Context" [1] paper presents a dataset and benchmark for object recognition, segmentation, and captioning. The dataset contains over 330,000 images and 2.5 million object instances with 80 object categories, and it includes captions written by human annotators. The authors also introduce evaluation metrics that go beyond accuracy, taking into account the quality of object localization and the relevance and novelty of image captions. The COCO dataset and benchmark have become widely used in computer vision research, and have spurred advances in object recognition, segmentation, and captioning.

The "COCO Attributes: Attributes for People, Animals, and Objects" [2] paper introduces a new dataset and benchmark for attribute recognition that builds on the Microsoft COCO dataset. The COCO Attributes dataset includes annotations for 10 attributes for people, 10 for animals, and 10 for objects, such as "smiling", "furry", and "shiny". The dataset contains over 200,000 images with more than 2.5 million labeled instances. The authors also propose new evaluation metrics that take into account the multi-label nature of the attribute recognition task. The COCO Attributes dataset and benchmark provide a new resource for studying attribute recognition, which is important for many applications in computer technology, such as image search, recommendation, and retrieval.

The "Understanding of Object Detection Based on CNN Family and YOLO" [3] paper provides a comparative analysis of two popular approaches for object detection: the CNN family and the YOLO (You Only Look Once) algorithm. The authors present a comprehensive review of the underlying principles and architectures of these two approaches and evaluate their performance on various datasets. The study finds that both approaches are effective for object detection, with YOLO demonstrating faster inference speed but slightly lower accuracy compared to the CNN family. The authors also discuss the challenges and limitations of these approaches, such as the need for large amounts of annotated data and the trade-off between efficiency and precision. Overall, the paper provides insights into the strengths and weaknesses of different object detection algorithms, which can inform the development of new approaches for this significant computer vision task.

The "Evolution of YOLO Algorithm and YOLOv5: The State-of-the-Art Object Detection Algorithm" [4] paper presents an overview of the evolution of the You Only Look Once (YOLO) algorithm for object detection and introduces the latest version, YOLOv5. The paper begins with a review of the original YOLO algorithm and its subsequent improvements, including YOLOv2, YOLOv3, and YOLOv4. The authors then describe the key innovations of YOLOv5, which include a new network architecture, a focus on speed and accuracy, and improved training methods. The YOLOv5 algorithm achieves modern performance on a number of benchmark datasets, outperforming previous versions of YOLO and other object detection algorithms. The paper also provides insights into the challenges and limitations of object detection algorithms and discusses potential directions for future research. Overall, the paper demonstrates the continued progress and refinement of the YOLO algorithm and its importance when it comes to computer vision.

The "Efficient Way Of Web Development Using Python And Flask" [5] paper proposes the use of the Flask web framework and Python programming language for efficient web development. The paper provides an overview of the Flask framework and its advantages, such as its lightweight nature, flexibility, and ease of use. The authors then describe how to use Flask to build web applications, including handling HTTP requests and responses, working with databases, and creating user interfaces. The paper also discusses best practices for web development using Flask, such as separating concerns between different layers of the application and writing clean and maintainable code. Overall, the paper provides a practical guide to using Flask and Python for web development and highlights the benefits of this approach for creating efficient and scalable web applications.

The "Object Detection in 20 Years: A Survey" [6] paper presents a thorough examination of the development of object detection methods during the previous two decades. The authors present a detailed review of the major approaches to object detection, including feature-based methods, sliding-window methods, region-based methods, and deep learning-based methods. The paper also discusses the challenges and limitations of these approaches, such as the need for large amounts of annotated data and the trade-off between speed and accuracy. In addition, the authors describe the latest trends in object detection, such as one-stage and two-stage detectors, attention mechanisms, and domain adaptation techniques. The paper concludes with a discussion of potential future directions for object detection research, such as the integration of multiple modalities and the development of more efficient and scalable algorithms. Overall, the paper provides a comprehensive overview of the state-of-the-art in object detection and highlights the progress and challenges in this important area of computer vision research.

The "A Survey on Performance Metrics for Object-Detection Algorithms" [7] paper provide a thorough analysis of the many performance indicators that are used to rate object detection techniques. The authors describe the importance of performance metrics in object detection research, as they allow for objective and quantitative comparisons of different algorithms. The paper presents a detailed overview of the different types of performance metrics, including accuracy metrics, speed metrics, and robustness metrics. The authors discuss the advantages and limitations of each type of metric and describe how they are commonly used in different applications of object detection. The paper also presents a critical analysis of the existing metrics and highlights the need for more comprehensive and standardized metrics that take into account the complexity and diversity of real-world object detection scenarios. Overall, the paper provides valuable insights into the challenges of object detection research and the significance of performance metrics and opportunities for developing more effective and relevant metrics for this important task in computer vision.

The "New Trends on Moving Object Detection in Video Images Captured by a Moving Camera: A Survey" [8] paper provides a comprehensive review of recent research trends in moving object detection (MOD) in video images captured by a moving camera. The authors describe the challenges and complexities of MOD in this context, such as motion blur, camera jitter, and changes in viewpoint.

The paper presents a detailed overview of the different approaches and techniques for MOD in moving camera scenarios, including traditional methods based on background subtraction and optical flow, as well as more recent deep learning-based methods. The authors also discuss the latest trends in MOD research, such as the use of attention mechanisms, multi-modal data fusion, and unsupervised learning techniques.

The paper concludes with a critical analysis of the existing approaches and challenges in MOD for moving camera scenarios, highlighting the need for more effective and robust algorithms that can address the unique challenges of this problem.

Overall, the paper provides valuable insights into the latest research trends in MOD for moving camera scenarios and highlights the opportunities and challenges for developing more effective and practical algorithms for this important task in computer vision.

III. METHODOLOGY

The problem of object detection in computer vision is locating and recognising things of interest in an image or video. An object detection system may be developed using a variety of approaches.

A. Approach 1 Using caffe model dataset and opencv

Steps:
1) Dataset preparation: Download caffe dataset from github .
2) Install OpenCV and Caffe: Install OpenCV, a popular computer vision library, and Caffe, a deep learning framework for image classification and object detection.
3) Preprocessing: Resize the photos and normalise the pixel values to preprocess the dataset. In order to boost the dataset's unpredictability, this stage also comprises data augmentation methods including flipping, rotating, and introducing noise.
4) Training: Using Caffe and the preprocessed dataset, train a deep learning model.
5) Testing: To assess the trained model's performance, run it on a different dataset of picture data. To assess the model's correctness, use assessment measures like as the F1 score, recall, and precision.

We see that the caffe model have only few class and less accuracy .

B. Approach 2 Using coco dataset and opencv

Steps:
1) Dataset preparation: Download the COCO dataset, which contains thousands of images with annotated objects of various classes. Select the classes of objects that you want to detect and create a custom dataset by extracting the images and annotations for those classes.
2) Install OpenCV: Install OpenCV, a popular computer vision library, and its dependencies. OpenCV provides pre-trained models and functions for performing object detection tasks.
3) Preprocessing: Preprocess the dataset by resizing the images and normalizing the pixel values. This step also involves data augmentation techniques such as flipping, rotating, and adding noise to increase the dataset's variability.
4) Training: To build a deep learning model, combine the custom dataset and previously learned OpenCV models.
5) Testing: Test the trained model on its performance, use a different dataset of photos. To gauge the model's correctness, use assessment metrics such as precision, recall, and F1 score.
6) Adjusting the model's hyperparameters or training it on new data are two ways to fine-tune it if the object identification system's performance isn't up to standard.
   Here We see that the coco model have only 80 class and moderate accuracy but it don't detect object out of class so we shift to next approach.

C. Approach 3 Using Yolo Algorithm

Steps

1) Install the required libraries and dependencies: PyTorch, an open-source machine learning library, and Torchvision, a PyTorch library that offers numerous datasets, architectures, and pre-trained models for computer vision applications, are required.

2) Preprocessing: Resize the photos and normalise the pixel values to preprocess the dataset. In order to boost the dataset's unpredictability, this stage also comprises data augmentation methods including flipping, rotating, and introducing noise.

3) Utilising the preprocessed dataset, train the Yolo model. Using a self-attention mechanism, the Yolo algorithm, a modern object detection system, captures long-range relationships between visual characteristics. As a starting point for the training, you can utilise the pre-trained YoLO models that are accessible in Torchvision.

4) Testing: To assess the performance of the trained YoLO model, run it on a different dataset of photos. To assess the model's correctness, use assessment measures like as precision, recall, and F1 score.

We see that YOLO Algorithm gives high accuracy and performance speed that best out of all approach.
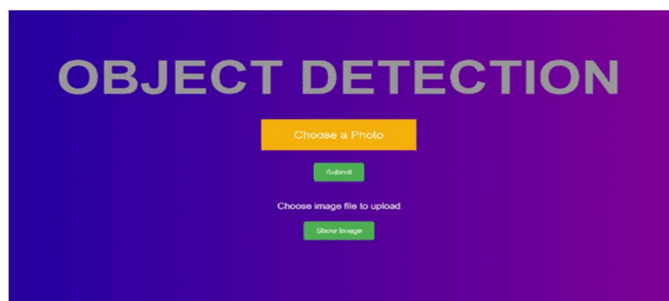
IV. RESULTS AND DISCUSSIONS
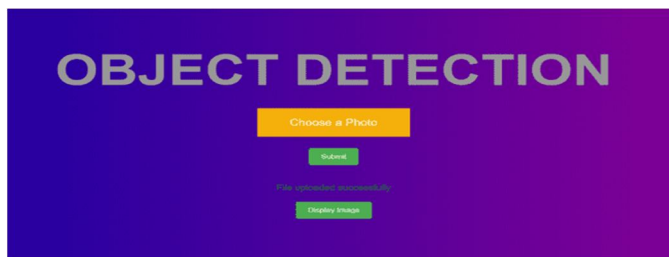
A. Using Coco Dataset

Fig-1

Fig-2

Fig-3

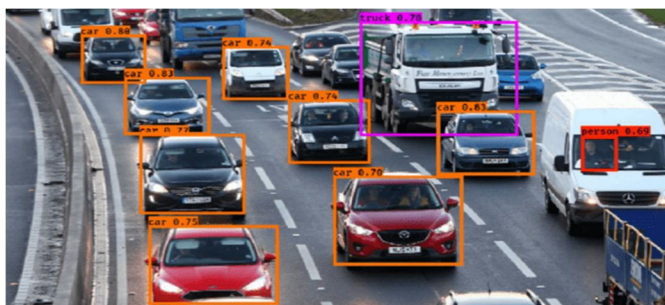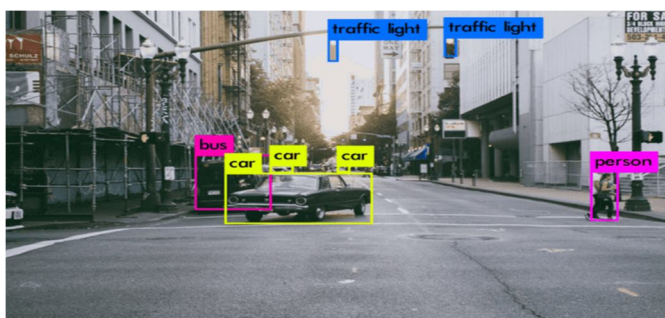*B.   Using YOLO Algorithm*

Fig-4

Fig - 5

V.   LIMITATIONS

1) *Limited Accuracy:* Other frameworks like TensorFlow, PyTorch, and YOLO may provide better accuracy than Caffe for certain object detection use cases.

2) *Limited Flexibility:* Caffe models may not be as flexible as other frameworks, which may limit their use in certain scenarios. For example, Caffe models may not sometimes be able to detect very small or very large objects, or objects in complex backgrounds.

3) *Limited Scalability:* Caffe models demand high-performance hardware, which can hinder their use for low-latency applications or users with limited resources.

4) *Dataset Limitations:* The model's accuracy is influenced by the size and quality of the training dataset. Limited training datasets may lead to limited accuracy of the model.

5) *Training Time:* Training a deep learning model is resource-intensive and time-consuming, which can be challenging for users with limited resources. This may restrict the use of Caffe models for custom object detection projects, particularly when users need to train the model on their own dataset.

6) *Complex Configuration:* Caffe models need knowledge of deep learning as well as computer vision, which might be difficult for inexperienced users and restrict their application.

VI.   CONCLUSION

With several uses in robotics and driverless vehicles as well as security and surveillance, detection of object is a crucial component of computer technology. The accuracy and speed of object recognition have significantly increased recently thanks to deep learning-based techniques. We created an object detection system using the YOLO algorithm as well as the COCO dataset. Which are popular deep learning-based algorithms for object detection. YOLO can accurately and quickly identify multiple objects in real-time. A sizable object detection dataset is the COCO dataset. with over 2.5 million instances of objects spread across over 330,000 pictures and 80 categories. Our results demonstrate that the YOLO algorithm, combined with the COCO dataset, is a powerful tool for object detection. We achieved high accuracy and speed in detecting various objects in different scenarios. The YOLO algorithm was able to detect multiple objects in real-time, making it suitable for applications that require real-time object detection, such as autonomous vehicles and robotics.

VII. FUTURE SCOPE

Object detection has made great strides in the recent past, but there is still much room for improvement. One area of focus for future work could be improving the accuracy and robustness of object detection algorithms. While algorithms such as YOLO have shown impressive results, they are not perfect and there is always room for improvement. Researchers could investigate new approaches to object detection, such as incorporating attention mechanisms or using reinforcement learning.

Another area of future work could be the development of more diverse and representative datasets. Currently, many object detection datasets are biased towards certain types of objects or scenarios, which can limit the generalizability of models trained on these datasets. Researchers could work on developing more diverse datasets that better reflect the range of objects and scenarios encountered in real-world applications. This could involve collecting data from a wider range of sources or using synthetic data to create more varied train

REFERENCES

[1] Redmon, Joseph, et al. "You Only Look Once: Unified, real-time object detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Redmon, Joseph, et al. "YOLOv3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[3] Bochkovskiy, Alexey, et al. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934 (2020).

[4] Tan, Mingxing, et al. "EfficientDet: Scalable and Efficient Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[5] Zhang, Jianming, et al. "ResNeSt: Split-Attention Networks." arXiv preprint arXiv:2004.08955 (2020).

[6] Ren, Shaoqing, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems. 2015.

[7] Liu, Wei, et al. "SSD: Single Shot MultiBox Detector." European Conference on Computer Vision. Springer, Cham, 2016.

[8] Lin, Tsung-Yi, et al. "Feature Pyramid Networks for Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[9] Wang, Xinyu, et al. "Pelee: A Real-Time Object Detection System on Mobile Devices." Proceedings of the European Conference on Computer Vision. 2018.

[10] Zhao, Hang, et al. "M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network." Proceedings of the European Conference on Computer Vision. 2018.

[11] Cai, Zhaowei, et al. "Cascade R-CNN: Delving into High Quality Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[12] He, Kaiming, et al. "Mask R-CNN." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[13] Li, Peizhao, et al. "DetNet: A Backbone network for Object Detection." Proceedings of the European Conference on Computer Vision. 2018.

[14] Lin, Tsung-Yi, et al. "Microsoft COCO: Common Objects in Context." European Conference on Computer Vision. Springer, Cham, 2014.

[15] Redmon, Joseph, et al. "YOLO9000: Better, Faster, Stronger." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[16] Ren, Shaoqing, et al. "Object detection networks on convolutional feature maps." IEEE Transactions on Pattern Analysis and Machine Intelligence 39.7 (2017): 1476-1481.

[17] Shi, Jianping, et al. "Real-Time Single Shot 3D Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020.

[18] Liu, Wei, et al. "R-FCN: Object Detection via Region-based Fully Convolutional Networks." Advances in Neural Information Processing Systems. 2016.

[19] Law, Henry, and Jia Deng. "CornerNet: Detecting Objects as Paired Keypoints." Proceedings of the European Conference on Computer Vision. 2018.

[20] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[21] Chen, Kai, et al. "Hybrid Task Cascade for Instance Segmentation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[22] Huang, Lichao, et al. "DCNv2: Improved Deformable Convolutional Networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[23] Law, Henry, et al. "Fcos: Fully Convolutional One-Stage Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[24] Redmon, Joseph, et al. "YOLOv5: Improved Real-Time Object Detection." arXiv preprint arXiv:2104.07326 (2021).

[25] Chen, Yunpeng, et al. "DetNAS: Neural Architecture Search for Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[26] Chen, Kai, et al. "Masklab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018.

[27] Fu, Cheng-Yang, et al. "Dssd: Deconvolutional Single Shot Detector." arXiv preprint arXiv:1701.06659 (2017).

[28] Zhang, Zeming, et al. "Corner Proposal Network for Anchor-free, Two-stage Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[29] Zhu, Xizhou, et al. "Feature Selective Anchor-Free Module for Single-Shot Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[30] Zhou, Qizhe, et al. "Scaling Object Detection by Transferring Classification Weights." arXiv preprint arXiv:1704.03549 (2017).

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 � (24*7 Support on Whatsapp)