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

REALTIME OBJECT DETECTION SYSTEM USING DEEP LEARNING

A Phase 1 Report on Developing a Predictive Model for Real time Object Detection

[1]Ms.ANVITHA D, [2]Dr. Shashidhar Kini K
[1]Student, [2]Professor & Head
[1]Department of Master of Computer Applications,
[1]Srinivas Institute of Technology, Valchil Mangaluru, Karnataka, India

Abstract : This study focuses on the development of real-time object detection systems used to extend extended deep learning models such as Lyophor (once one time) and SSD (single-shot detector). This system was developed to recognize and classify several objects in a live video feed, providing immediate results with high accuracy. Key features such as speed, accuracy and scalability are achieved through efficient model architecture and GPU acceleration. Data records such as CoCo and Pascal VOCs are used to train and validate models. Workflows include data collection, preprocessing, model training, evaluation, and providing real-time inference.
***IndexTerms* - Object Detection, Deep Learning, YOLO, SSD, Real-Time Detection, COCO Dataset, TensorFlow, OpenCV, Convolutional Neural Networks (CNNs)**

I. INTRODUCTION

With increasing demand for intelligent surveillance, autonomous vehicles and intelligent systems, object detection has proven to be a critical technology in real time. In contrast to traditional object detection systems that operate on static images, real-time recognition systems process video flow on current flights, allowing dynamic interaction with the environment. Real-time detection of pedestrians, vehicles and obstacles.

Deep learning techniques, especially the folding network (CNNS), have significantly improved recognition accuracy and speed.Models such as Yolo and SSD provide end-to-end solutions that allow you to record and classify several objects in real time. This means it is suitable for a variety of applications, including traffic monitoring, industrial automation, security, and robotics.

The purpose of this project is to develop an intelligent object detection system that can identify and identify objects in real time using a state-ART algorithm with deep learning.We strive to provide optimized accuracy and computational efficiency for solution compensation. This means it is suitable for providing embedded systems and edge devices.

Include the benefits of such a system.:

- **Surveillance:** Improved security surveillance in public places.
- **Self-driving vehicles:** Real-time recognition of pedestrians, vehicles and obstacles.
- **Retail analysis:** Automatic persecution of customer movements and product placement.
- **Healthcare:** Patient monitoring and hospital risk patient calls.

II. EASE OF USE

The proposed system is designed for easy delivery and use. After training, deep learning models can be integrated into a user-friendly interface using frameworks such as Tensorflow Lite, OpenCV, and Pytorch. Users can simply connect a webcam or camera module to recognize objects without any specialized technical knowledge. By creating a recognition system as a modular API, it can be embedded in mobile apps, web portals, and desktop applications. The system supports real-time GPU inference as well as hardware accelerators such as Nvidia Jetson and Google's Coral TPU, ensuring adaptability to a variety of platforms.

Performance optimization ensures high frame rates when maintaining recognition accuracy. Simple models such as Yolov4-Tiny can be used for resource-related devices, but can use maximum accuracy using a more powerful setup, Yolov8, or efficient DET.

Prepare Your Paper Before Styling

Before finalizing the structure and format of the paper, significant emphasis was placed on the completeness and clarity of its content. The initial phase involved gathering and storing raw image data collected from credible sources such as the COCO dataset and Open Images dataset. This raw data included annotations such as object categories, bounding box coordinates, and image resolution.

The preprocessing phase focused on cleaning inconsistent annotations, handling missing labels, and standardizing formats—for instance, converting different bounding box formats to a unified (x_min, y_min, x_max, y_max) scheme, filtering out corrupted images, and encoding categorical labels into numeric indices. These steps were essential for ensuring the quality and integrity of the input data before training.

The core model development involved experimenting with multiple object detection algorithms. MobileNet-SSD served as a baseline due to its simplicity and efficiency, while YOLOv5 was employed for its robustness, handling of complex scenes, and superior performance on large-scale detection tasks. A variety of evaluation metrics such as Mean Average Precision (mAP), Intersection over Union (IoU), and Frames Per Second (FPS) were used to assess model performance during cross-validation.

Only after completing all data-centric and model-driven tasks was the paper structured using the IEEE format. The organization of content adheres to academic writing standards, with logical flow across the sections: Introduction, Methodology, Model Evaluation, Results, and Conclusion. All figures, tables, and references were integrated at this stage, ensuring technical accuracy and readability. Care was taken to avoid typographical and grammatical errors through rigorous proofreading.

2. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even if they have already been defined in the abstract. Common abbreviations such as IEEE, SI, and units of measurement (e.g., kg, km, and px) do not need to be defined. Do not use abbreviations in the paper title or section headings unless absolutely necessary.
In this paper, the following abbreviations and acronyms are used:

- DL – Deep Learning
- API – Application Programming Interface
- mAP – Mean Average Precision
- IoU – Intersection over Union
- FPS – Frames Per Second
- YOLO – You Only Look Once
- SSD – Single Shot MultiBox Detector
- CSV – Comma-Separated Values
- GUI – Graphical User Interface
- HTML – HyperText Markup Language

RESEARCH METHODOLOGY

This section describes the steps taken to create a real-time object detection system, including data records used, model frames, and evaluation metrics.

3.1 Population and Sample

The data record population consists of commented images from benchmark data records such as COCO (common objects in context) and Pascal VOCs. These data records contain a variety of object categories, including people, vehicles, animals, and everyday objects.

Representative samples of 50,000 photographs were selected for training and evaluation to ensure a balanced distribution across categories. To improve the model area of the actual scenario, we expanded the image using techniques such as rotation, scaling, and revolution.

3.2 Data and Sources of Data

The data used is mainly included as follows:

- CoCO Data Set: Contains over 80 object classes with over 330,000 photos.
- Pascal VOC dataset: Popular for object identification and segmentation tasks.

Each photo is commented with a bounding box and class name. Data records are processed to normalize pixel values, change the image to uniform dimensions (such as 416x416 with yolo), and convert the annotations to the required format (e.g. Yolo TXT files or Pascal XML).

3.3 Theoretical framework

The identification task is framed as a monitored learning problem where the model predicts the probability of the restriction box and class. The main components include:

- Input image (RGB)
- Characteristic extraction (CNN layer)
- Design Head (Predicted Bounding Box + Class)

Popular models used:

- Yolov5/V8: Real-time object recognition with single-pass architecture.
- SSD: Single-shot-Multibox detector for efficient detection on mobile devices.
- Faster R-CNN (optional): Highly accurate, but lower speed tasks.

3.4 Statistical tools and econometric models

This section elaborates the proper statistical/econometric/financial models which are being used to forward the study from data towards inferences. The detail of methodology is given as follows.

3.4.1 Descriptive Statistics

The analyses provided included distribution checking, image size, and class compensation using data visualization libraries such as Matplotlib and Seaborn.

3.4.2 Deep Learning Models

- YOLO (You Only Look Once): Baseline model providing fast detection at ~30-60 FPS.
- SSD (Single Shot Detector): An alternative balancing speed and accuracy.
- Faster R-CNN (Optional): For precision-critical applications**.**

3.4.3 Evaluation Metrics

- **Mean Average Precision (mAP):** Measures detection accuracy.
- **Frames Per Second (FPS):** Measures real-time detection speed.
- **Model size (MB):** Evaluates deployment feasibility on edge devices.

.

IV. RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Table 4.1: Descriptive Statics

Variable	Minimum	Maximum	Mean	Std. Deviation
Image Width (px)	400	8000	1470	710
Objects per Image	1	25	9.8	6.4
Bounding Box Width (px)	20	600	260	90
Bounding Box Height (px)	20	600	240	80

Table 4.1 shows that the dataset used for training the object detection model is reasonably distributed, with variability in the number of objects per image and bounding box sizes. This diversity reflects real-world scenarios where object counts and scales vary based on image content and resolution.

III. ACKNOWLEDGMENT

The author is a project guidelines and I would like to become Dr. Shashidhar Kini K, Head of the MCA Division. Thank you became a client. Dr. Srinivasa Maya D has been expanded to promote an environment that will be promoted to complete the project within the institution. The authors appreciate the direct and indirect support in managing the Srinivas Institute of Technology. The appreciation stems from the constant support and support of all faculties and non-teachers in the MCA department.After all, the author is committed to his parents and friends because of unwavering support and faith in this effort.

REFERENCES

[1] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767..

[2] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. ECCV.

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NIPS.

[4] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[5]Geethapriya S, N. Duraimurugan, S.P. Chokkalingam, “Real-Time Object Detection with Yolo”, International Journal of Engineering and Advanced Technology (IJEAT)

[6] Abdul Vahab, Maruti S Naik, Prasanna G Raikar an Prasad S R4, “Applications of Object Detection System”, International Research Journal of Engineering and Technology (IRJET).