Sentinel – Scene Analysis using DETR Transform Model

G. Pranavi
Assistant Professor
Dept. of Computer Science and Engineering Jyothishmathi Institute of Technology and Science (JNTUH)
Karimnagar, Telangana, India
gunda.pranavi@gmail.com

Mandal Nikhitha
UG Student
Dept. of Computer Science and Engineering Jyothishmathi Institute of Technology and Science (JNTUH)
Karimnagar, Telangana, India
226684nikitha@gmail.com

B. Harish
UG Student
Dept. of Computer Science and Engineering Jyothishmathi Institute of Technology and Science (JNTUH)
Karimnagar, Telangana, India
bhukyaharish903@gmail.com

B. Keerthana
UG Student
Dept. of Computer Science and Engineering Jyothishmathi Institute of Technology and Science (JNTUH)
Karimnagar, Telangana, India
keerthanaboga4@gmail.com

V. Saketh
UG Student
Dept. of Computer Science and Engineering Jyothishmathi Institute of Technology and Science (JNTUH)
Karimnagar, Telangana, India
sakethvasam8@gmail.com

***Abstract*—**This project presents a deep learning–based sys- tem for real-time object detection in dynamic environments, developed using the DETR (Detection Transformer) model and implemented with Streamlit as the frontend interface. The system accepts input from both image files and live camera streams, enabling accurate object detection in static images as well as continuous video processing. The architecture integrates ResNet-50 as the backbone network for feature extraction, providing robust visual feature representations prior to transformer-based analysis.

In the proposed framework, input images or video frames are first processed through ResNet-50 to extract high-level feature maps. These feature representations are enhanced with positional encoding and passed into the transformer encoder–decoder structure of DETR. The encoder captures global contextual dependencies using multi-head self-attention mechanisms, while the decoder predicts a fixed set of object queries corresponding to class labels and bounding box coordinates. Unlike conventional object detection approaches that rely on anchor boxes and non-maximum suppression, the proposed deep learning model formulates detection as a set prediction problem and employs bipartite matching loss to ensure accurate one-to-one object correspondence.

The system is deployed with a Streamlit-based frontend to provide an interactive and user-friendly interface for uploading images, processing video streams, and visualizing detection results in real time. Experimental evaluation demonstrates that the integration of deep convolutional feature extraction, transformer- based global reasoning, and an interactive frontend provides an efficient, scalable, and practical solution for real-time environ- mental object detection applications.

***Index Terms*—**Deep Learning, DETR, ResNet-50, Real-Time Object Detection, Video Processing, Streamlit, Transformer.

I. Introduction

Real-time object detection plays a crucial role in modern intelligent vision systems used for surveillance, traffic monitoring, environmental observation, and security applications. Traditional computer vision methods relied on handcrafted features and complex multi-stage pipelines, which required anchor boxes and non-maximum suppression. These approaches often lacked global contextual understanding and scalability in dynamic environments.

With the advancement of deep learning, convolutional neu- ral networks significantly improved object detection accu- racy by automatically learning hierarchical feature represen- tations. However, many CNN-based models still depend on region proposals and post-processing steps. To overcome these limitations, transformer-based architectures were introduced into computer vision. The DETR (Detection Transformer) reformulates object detection as a direct set prediction prob- lem, eliminating anchor boxes and non-maximum suppression while enabling global contextual reasoning through attention mechanisms.

The proposed project, Sentinel, is a deep learning–based real-time object detection system developed using DETR with ResNet-50 as the backbone network for feature extraction. The system processes both static images and live video streams captured from cameras. ResNet-50 extracts high-level visual features, which are then passed to the transformer encoder–decoder architecture for accurate object classification and bounding box prediction. The self-attention mechanism allows the model to understand relationships between multiple objects within a scene.

The system is deployed using Streamlit as the frontend interface, providing an interactive platform for uploading images, streaming video, and visualizing detection results in real time.

II. Literature Review

A. DETR – End-to-End Object Detection with Transformers (Carion et al., 2020)

Carion et al. (2020) introduced the DETR (Detection Transformer), a transformer-based object detection framework that reformulates detection as a direct set prediction problem. Unlike traditional detection models, DETR eliminates anchor boxes and Non-Maximum Suppression (NMS) by using bipartite matching loss for one-to-one object assignment. The model integrates a convolutional backbone such as ResNet-50 with a transformer encoder–decoder architecture to capture global contextual relationships through self-attention mechanisms. DETR simplifies the detection pipeline while achieving competitive performance on benchmark datasets.

B. Faster R-CNN – Towards Real-Time Object Detection (Ren et al., 2015)

Ren et al. (2015) proposed Faster R-CNN, which introduced the Region Proposal Network (RPN) to improve the efficiency of region-based object detection. This model significantly enhanced detection accuracy and speed compared to earlier R-CNN versions. Faster R-CNN enabled nearly end-to-end training but still relied on anchor boxes and post-processing techniques such as NMS. Although highly accurate, it was computationally intensive for real-time applications.

C. YOLOv4 – Optimal Speed and Accuracy of Object Detection (Bochkovskiy et al., 2020)

Bochkovskiy et al. (2020) introduced YOLOv4, a single-stage detector optimized for both speed and accuracy. YOLOv4 incorporated CSPDarknet53 as its backbone and applied various optimization strategies to improve performance without increasing inference cost. The model achieved real- time object detection with impressive accuracy. However, it continued to depend on anchor-based detection mechanisms and grid-based predictions.

D. Deformable DETR – Deformable Transformers for End-to-End Object Detection (Zhu et al., 2021)

Zhu et al. (2021) proposed Deformable DETR to address the slow convergence and computational complexity of the original DETR model. The authors introduced deformable attention modules that focus on a limited number of key sampling points instead of the entire feature map. This approach reduced training time and improved detection performance, particularly for small objects, while preserving the end-to-end transformer-based framework.

E. YOLOv7 – Trainable Bag-of-Freebies Sets New State-of-the-Art (Wang et al., 2022)

Wang et al. (2022) introduced YOLOv7, which further improved real-time object detection through architectural enhancements and advanced training techniques referred to as ”trainable bag-of-freebies.” YOLOv7 achieved state-of-the-art performance in both speed and accuracy among real-time detectors. Despite its efficiency, it still followed an anchor-based detection approach and required post-processing steps.

III. Methodology

The proposed system is designed to perform real-time object detection in dynamic environments using a deep learning–based framework. The methodology integrates the DETR (De- tection Transformer) model with ResNet-50 as the backbone network for feature extraction and Streamlit as the frontend interface. The overall methodology is structured into the following stages:

A. Data Acquisition

The system accepts input from two sources: static image files and live camera streams. For video processing, the live stream is segmented into sequential frames, and each frame is processed independently for object detection. This enables both image-based detection and real-time video analysis.

B. Data Preprocessing

The acquired images or video frames undergo preprocessing to ensure compatibility with the deep learning model. The preprocessing steps include resizing the input to the required dimensions, normalizing pixel values, and converting the data into tensor format. These steps enhance model stability and improve detection accuracy.

C. Feature Extraction using ResNet-50

The preprocessed input is passed through the ResNet-50 convolutional neural network backbone. ResNet-50 utilizes residual learning to extract deep hierarchical feature represen- tations from the input data. These feature maps capture spatial and semantic information essential for object detection.

D. Transformer-Based Detection using DETR

The extracted feature maps are enriched with positional encoding and fed into the transformer encoder–decoder archi- tecture of DETR. The encoder applies multi-head self-attention mechanisms to capture global contextual relationships between objects within the scene. The decoder processes a fixed set of learned object queries to predict object class labels and bounding box coordinates. DETR formulates object detec- tion as a direct set prediction problem and applies bipartite matching loss to ensure unique, one-to-one object assignment, eliminating the need for anchor boxes and Non-Maximum Suppression (NMS).

E. Output Generation and Visualization

The model outputs object class labels, bounding box coordinates, and corresponding confidence scores. For video input, bounding boxes are drawn on each processed frame in real time. The final detection results are displayed through a Streamlit-based frontend interface, enabling users to upload images, stream video, and visualize detection outputs interactively.

The proposed methodology combines deep convolutional feature extraction with transformer-based global reasoning to achieve accurate and efficient real-time object detection. The integration of image and video processing capabilities with an interactive frontend ensures scalability, usability, and practical deployment in real-world environments.

IV. Tools and Libraries

The proposed real-time object detection system is developed using various software tools and deep learning libraries to ensure efficient implementation and deployment. The following tools and libraries are utilized in the project:

A. Programming Language

Python – The primary programming language used for implementing the deep learning model, backend processing, and frontend integration due to its extensive support for machine learning and computer vision libraries.

B. Deep Learning Frameworks

PyTorch – Used for implementing and loading the DETR model and managing neural network operations such as training, inference, and tensor computations.

Torchvision – Provides pre-trained models including ResNet-50 and image transformation utilities for preprocessing tasks.

C. Computer Vision Libraries

OpenCV – Used for handling image processing, video frame extraction, camera streaming, and drawing bounding boxes on detected objects.

Pillow (PIL) – Used for image loading and basic image manipulation.

D. Transformer and Model Utilities

Hugging Face Transformers – Provides access to pre-trained transformer models and utilities for loading and fine-tuning DETR architecture.

E. Frontend Development

Streamlit – Used to develop an interactive web-based frontend interface. It enables image uploads, live camera streaming, and real-time visualization of object detection results without complex web development requirements.

F. Development Environment

Visual Studio Code (VS Code) – Used as the primary code editor for development.

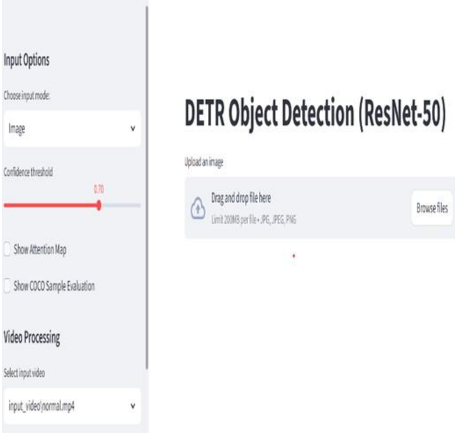
Anaconda / Virtual Environment – Used for managing dependencies and Python environments.

G. Hardware Requirements

GPU (Optional but Recommended) – NVIDIA GPU for faster inference and real-time performance.

CPU-Based System – The model can also run on CPU for basic detection tasks, though with reduced speed.

V. Results



DETR Object Detection (ResNet-50)

Input Options

Choose input mode: Image

Confidence threshold 0.70

□ Show Attention Map

□ Show COCO Sample Evaluation

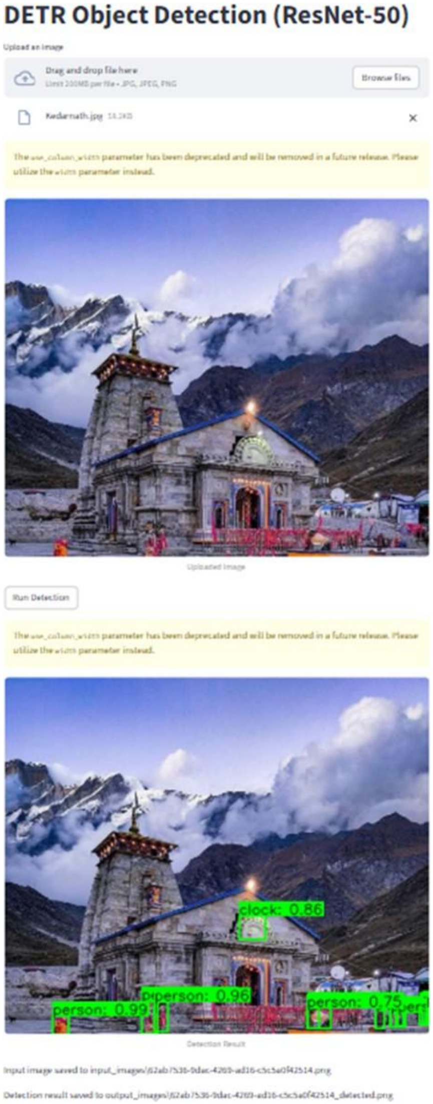
Video Processing

Select input video: input_video\normal.mp4

Upload an image

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Fig. 1: Dashboard for Input



DETR Object Detection (ResNet-50)

Upload an image

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Kedarnath.jpg

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the width parameter instead.

Uploaded Image

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the width parameter instead.

Detection Result

Input image saved to input_images\62ab7536-9dac-4269-ad16-c5c5a0f42514.png

Detection result saved to output_images\62ab7536-9dac-4269-ad16-c5c5a0f42514_detected.png

Fig. 2: Detection output with multiple objects identified in the scene

VI. Discussions

A. Model Performance

The proposed system utilizes the DETR architecture with ResNet-50 as the backbone for feature extraction. The transformer-based attention mechanism enables global contextual understanding, resulting in accurate object classification and bounding box prediction. The end-to-end detection framework eliminates anchor boxes and non-maximum suppression, reducing pipeline complexity and improving detection consistency in complex scenes.

B. Real-Time Processing Capability

The system supports both static image detection and live video stream processing from cameras. Real-time inference performance is satisfactory on GPU-enabled systems, while CPU-based execution provides moderate speed suitable for small-scale applications. The system maintains detection stability across continuous video frames.

C. Practical Applicability

The developed framework can be applied to surveillance, traffic monitoring, and environmental observation systems. The Streamlit-based frontend provides an interactive interface for uploading images and streaming live video, making the system user-friendly and easily deployable.

D. Limitations

Transformer-based models require significant computational resources. Detection accuracy may vary under poor lighting conditions, occlusion, or very small object sizes. Real-time performance may decrease without hardware acceleration.

E. Future Enhancements

Future improvements may include model optimization, lightweight transformer variants, integration of object tracking algorithms, and deployment on edge devices for enhanced scalability and efficiency.

VII. Conclusion

This paper presented a deep learning–based real-time object detection system for dynamic environmental monitoring using the DETR architecture with ResNet-50 as the backbone for feature extraction. The proposed framework combines convolutional neural networks and transformer-based attention mechanisms to achieve accurate object classification and precise bounding box localization. By formulating detection as a direct set prediction problem, the system eliminates the need for anchor boxes and non-maximum suppression, thereby simplifying the detection pipeline while improving contextual understanding.

The system supports both image-based detection and real-time video processing through camera input and is deployed using a Streamlit-based frontend for interactive visualization. Experimental results demonstrate that the model provides efficient and scalable performance in real-world environments. Overall, the integration of deep feature extraction and transformer-based reasoning offers a robust and practical solution for intelligent real-time object detection applications.

VIII. References

References

[1] YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *arXiv*, 2022. Available: https://arxiv.org/abs/2207.02696

[2] J. Redmon et al., ”You Only Look Once: Unified, Real-Time Object Detection,” *arXiv*, 2015. Available: https://arxiv.org/abs/1506.02640

[3] A. Wang et al., ”YOLOv10: Real-Time End-to-End Object Detection,” *arXiv*, 2024. Available: https://arxiv.org/abs/2405.14458

[4] ”DETRs Beat YOLOs on Real-time Object Detection,” *Liner*. Available: https://liner.com/review/detrs-beat-yolos-on-realtime-object-detection

[5] ”Improved object detection method for unmanned driving based on Transformers,” *Frontiers*, 2024. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2024.1342126/full

[6] ”Object detection using convolutional neural networks and transformer-based models: a review,” *Journal of Electrical Systems and Information Technology*, 2023. Available: https://jesit.springeropen.com/articles/10.1186/s43067-023-00123-z

[7] ”A Comparative Analysis of State-of-the-Art Object Detection Models,” IoT Digital Twin PLM, 2025. Available: https://iotdigitaltwinplm.com/ a-comparative-analysis-of-state-of-the-art-object-detection-models/

[8] [8] ”Object detection survey for industrial applications with focus on quality control,” *Springer Nature*, 2025. Available: https://link.springer.com/article/10.1007/s11740-025-01369-4

[9] ”RT-DETRs Beat YOLOs on Real-Time Object Detection (CVPR 2024),” *CVPR Open Access*. Available: https://openaccess.thecvf.com/content/CVPR2024/papers/Zhao DETRs Beat _YOLOs _ on Real-time Object Detection CVPR 2024 paper.pdf

[10] ”Best Object Detection Models 2025: RF-DETR, YOLOv12 & Beyond,” *Roboflow Blog*, 2025. Available: https://blog.roboflow.com/ best-object-detection-models/