OBJECT RECOGNITION

Simran Choudhary
IIITD
simran21205@iiitd.ac.in

Siddhant
IIITD
siddhant21204@iiitd.ac.in

Aishiki Bhatacharya
IIITD
aishiki21007@iiitd.ac.in

Abstract

This study assesses the performance of two prominent object recognition models, YOLOv5 and Faster R-CNN, using the Indian Driving Dataset (IDD), which features complex and dynamic driving environments typical of Indian road conditions. The objective is to compare these models in terms of their accuracy, speed, and reliability in detecting various objects such as vehicles, pedestrians, and road signs amidst the challenging scenes presented in the IDD. Following the performance benchmarking, we employ the Tool for Identifying Errors in Detections (TIDE) to conduct a detailed error analysis. The findings aim to provide insights into the strengths and limitations of each model, offering guidance on their applicability in real-world, diverse driving contexts like those found in India.

1. Introduction

Object recognition is pivotal in computer vision for developing autonomous driving systems and advanced driver-assistance systems (ADAS). These technologies must accurately detect and interpret various objects in real time to ensure the safety and efficiency of vehicle operations. However, the performance of object recognition models can vary significantly depending on the dataset used for training and evaluation, especially in diverse and unpredictable environments.

The Indian Driving Dataset (IDD) provides a comprehensive environment that captures the complex dynamics of driving in India, characterized by dense traffic, a variety of vehicle types, frequent pedestrian crossings, and non-standard road infrastructures. This dataset is particularly challenging due to its diverse conditions, including urban congestion and rural sparsity, making it an ideal benchmark for testing the robustness of object recognition models.

This project aims to compare two advanced object recognition models: YOLOv5 and Faster R-CNN. These models are selected for their widespread use and proven efficiency in various object detection tasks. We will evaluate their performance on the IDD to understand their effectiveness in detecting objects in challenging and cluttered road scenes typical of Indian roads.

Following the performance evaluation, a detailed error analysis will be conducted using the Tool for Identifying Errors in Detections (TIDE). This analysis will help identify the specific error types that each model is prone to in this unique dataset, including the conditions under which these errors are most likely to occur.

Through this study, we seek to elucidate the strengths and weaknesses of each model, providing insights into their practical applicability and guiding future improvements to enhance the reliability of object recognition systems in similar challenging environments.

2. Methodology

2.1. YOLOv5

The YOLOv5 [2]model is adept at high-speed and accurate object detection, utilizing a streamlined architecture for real-time image processing. The following subsections detail the components of this architecture.

Backbone : The backbone is the primary feature extraction component of YOLOv5. It processes the input image to extract hierarchical feature maps at various scales. These maps are essential for detecting objects of different sizes within the image.

Neck (Feature Fusion Network) : The feature fusion network, or the neck, enhances the object detection capabilities of YOLOv5. It integrates the multi-scale feature maps from the backbone to produce three different sizes of feature maps: P3 (80x80), P4 (40x40), and P5 (20x20). These maps correspond to the detection of small, medium, and large objects, respectively.

Prediction Heads : The processed feature maps are forwarded to the prediction heads, where two main operations occur: confidence calculation and bounding-box regression. Using predefined anchor boxes, the model calculates class probabilities, objectness scores, and coordinates for bounding boxes, which are stored in a multi-dimensional array containing details such as object class, confidence level, and box dimensions.

Post-Processing : The model applies thresholds—confidence threshold (*confthreshold*) and objectness threshold (*objthreshold*)—to filter out detections with low probability. Additionally, a non-maximum suppression (NMS) process is executed to eliminate overlapping bounding boxes, ensuring that only the most probable box for each detected object is retained.

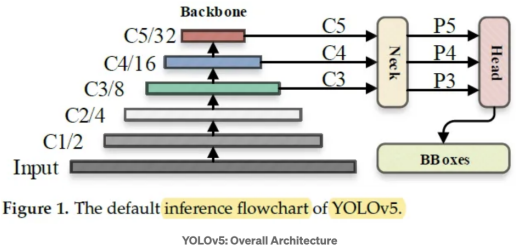

Figure 1. YOLOv5 Architecture

2.2. Faster RCNN

Region Proposal Network (RPN) : For anchors, we use three scales with box areas of 128^2, 256^2, and 512^2 pixels, and three aspect ratios of 1:1, 1:2, and 2:1. By default, we use three scales and three aspect ratios, yielding $k = 9$ anchors at each sliding position. [3]For a convolutional feature map of size $W \times H$ (typically ≈ 2400), there are $W \times H \times k$ anchors in total.

ROI Pooling : ROI Pooling splits the input feature map into a fixed number (let's say k) of roughly equal regions, and then applies Max-Pooling on every region. Therefore, the output of ROI Pooling is always k, regardless of the size of the input.

Loss function : Localization Loss (reg) for the region proposal network using Smooth L1 loss, which regresses the locations of the proposed regions towards the ground truth object locations.
Objectness Loss (cls) uses cross-entropy loss, which determines whether each anchor should be considered to contain an object or not.

3. Dataset Description

3.1. Indian Driving Dataset (IDD)

The Indian Driving Dataset (IDD) [4] is specifically designed to explore the challenges of autonomous navigation in unstructured environments, particularly those that are common in countries with diverse and chaotic traffic conditions like India. Unlike most datasets that focus on structured driving environments with clear lane markings and predictable vehicle behavior, IDD captures the intricacies and unpredictability of Indian roads.

Dataset Characteristics IDD comprises 10,004 images collected from 182 driving sequences across various locations in India. These images are finely annotated with 34 different classes, reflecting a broad spectrum of road participants and scenarios. This dataset includes several unique classes not typically found in other datasets like Cityscapes, such as autorickshaws and animals, alongside more common classes such as vehicles and pedestrians.

3.2. KITTI

The object detection and object orientation estimation benchmark consists of 7481 training images and 7518 test images, comprising a total of 80.256 labelled objects. All images are coloured and saved as PNGs. Useful labels for evaluation in KITTI 2d include car, van, truck and pedestrian.

3.3. UKNEC

The UA-DETRAC benchmark dataset consists of 100 challenging videos captured from real-world traffic scenes (over 140, 000 frames with rich annotations, including illumination, vehicle type, occlusion, truncation ratio, and vehicle bounding boxes) for multi-object detection and tracking.

4. Results

Analysis of YOLOv5 on IDD The results of deploying the YOLOv5 model on the Indian Driving Dataset (IDD) Test set is displayed below. Each frame within the image illustrates multiple object detections made by the model, highlighted by colored bounding boxes that encapsulate various entities such as vehicles, people, and traffic signs. The labels associated with each bounding box identify the class of the detected object, such as 'car', 'bus', 'motorcycle', 'traffic sign', etc.

The image showcases the model's ability to recognize and localize multiple object types in diverse and dynamic traffic scenes typical of Indian roads. Each detection is marked by a bounding box with a specific color and is accompanied by a label indicating the detected object type.

This visualization aids in assessing the model's performance in terms of its precision and recall across different classes.

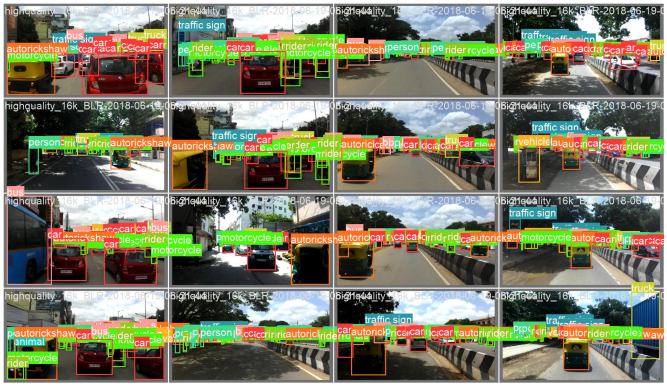

Figure 2. YOLOv5 Output

Class	AP
Car	65.9
Bus	69.3
autorickshaw	68
truck	61.7
Person	49.3
Bicycle	39.7
train	24.9
animal	24.9
rider	51.8

Table 1. AP values for different classes in the IDD using YOLOv5

The table displays the Average Precision (AP) scores for various classes detected by an object recognition model on the Indian Driving Dataset (IDD). The model performs best in detecting larger vehicles, with buses achieving the highest AP at 69.3 and cars at 65.9, indicating robust detection capabilities for these vehicle types. Autorickshaws and trucks also show high precision, with AP scores of 68 and 61.7, respectively. However, the model struggles with smaller or less distinct objects such as bicycles, trains, and animals, all scoring below 40 AP. The detection of humans (persons) and riders present moderate challenges, with APs of 49.3 and 51.8, suggesting areas where further model tuning and enriched training data might be beneficial to enhance accuracy and reduce misclassification.

Analysis of Fast RCNN on KITTI 2d The AP values for various classes using Faster R-CNN on the KITTI dataset reveal distinct insights: High performance for Pedestrian, Tram, and Car categories indicates strong detection capabilities in traffic scenes. However, moderate AP values for

Figure 3. Faster RCNN outputs

Class	AP
Car	74.06
Van	70.89
Truck	74.82
Pedestrian	79.20
Person_sitting	70.41
Cyclist	70.41
Tram	79.48
Misc	75.42
DontCare	69.23

Table 2. AP values for different classes in the KITTI dataset using Faster R-CNN

Van, Truck, Person sitting, and Misc suggest potential improvements in feature recognition and handling occlusions. The overall variation in detection performance underscores the importance of robust detection algorithms for enhancing the efficacy of multi-object tracking systems.

5. TIDE Analysis

TIDE[1] is a novel framework designed to assess error sources in object detection and instance segmentation algorithms. It operates universally across datasets and directly analyzes prediction files, offering a detailed alternative to traditional mAP computations. The framework categorizes errors into six types and quantifies their impact on performance.

On YOLOv5 The TIDE analysis on the YOLOv5 model using the UA-DETRAC dataset highlights various error types impacting model performance. Notably, localization errors (LocError) are the most frequent, constituting 32.4%

of all errors. Classification errors (ClassError) and background confusions (BackgroundError) are less common, contributing 9.7% and 9.2% of the errors. Understanding the distribution of these errors is crucial for pinpointing specific weaknesses in the model's object detection capabilities.

The impact of these errors on model accuracy varies, with localization errors being the most detrimental, having an impact score of approximately 0.6. This indicates a significant reduction in accuracy due to poorly localized bounding boxes. Bounding box errors also greatly impact performance, scoring around 0.5. The least impactful errors are related to background confusion, with a score just above 0.2. This analysis suggests that localization and bounding box accuracy improvements could yield substantial gains in the overall model performance.

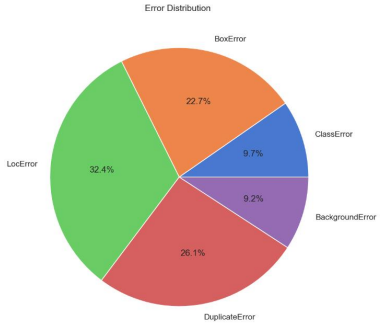

Figure 4. Summary plot of YOLOv5's errors

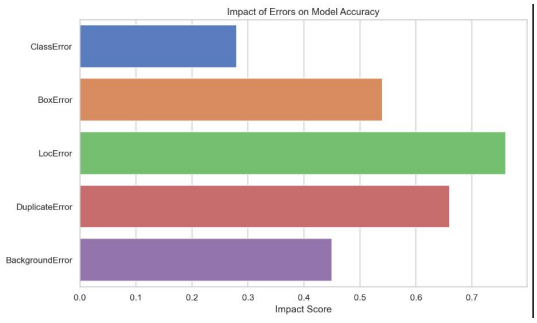

Figure 5. Summary plot of YOLOv5's errors

On Faster R-CNN The TIDE analysis applied to the Faster R-CNN model on the UA-DETRAC dataset reveals a diverse distribution of error types affecting model performance. The most prevalent errors are BoxError, representing 28.4% of total errors, indicating issues with bounding box precision. Localization errors (LocError) and BackgroundError are less frequent, constituting 14.2% and 12.8%, respectively. These statistics are essential for understanding specific areas where the Faster R-CNN model may require optimization.

In terms of their impact on model accuracy, BoxError presents the most significant challenge, with an impact score of approximately 0.4. This suggests substantial accuracy degradation due to incorrect or imprecise bounding box predictions. ClassError and LocError also substantially affect the model's accuracy, highlighting the critical need for classification and localization precision improvements. On the other hand, DuplicateError and BackgroundError show a comparatively lower impact on model performance, with impact scores closer to 0.2. Addressing these higher-impact errors could lead to notable improvements in the model's overall effectiveness.

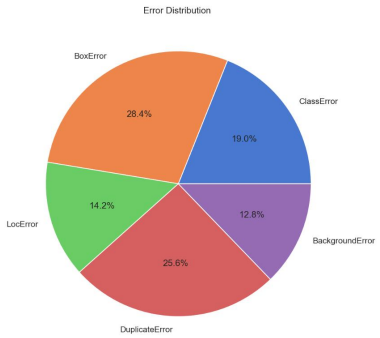

Figure 6. Summary plot of Faster RCNN's errors

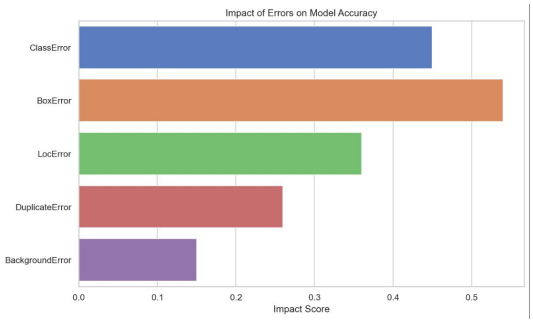

Figure 7. Summary plot of Faster RCNN's errors

6. Conclusion

The analysis of YOLOv5 on the Indian Driving Dataset (IDD) and Faster R-CNN on the KITTI dataset underscores the strength and limitations of object detection models in varying contexts. While YOLOv5 shows strong performance for larger vehicles on IDD but struggles with smaller objects, Faster R-CNN excels with traffic-related objects on KITTI, though it could improve in detecting vans and trucks. This comparison highlights the necessity for ongoing model enhancements and training data diversification to

boost accuracy and adaptability in diverse tracking environments.

Team Member Commentaries

Simran Choudhary

In an evaluation conducted using the Indian Driving Dataset (IDD), this study compared the effectiveness of two object recognition models: YOLOv5 and Faster R-CNN. The primary focus was on assessing their accuracy and processing speed. Detailed error analysis was facilitated by utilizing TIDE, a specialized software tool.

YOLOv5 has shown good processing speed, making it a suitable choice for real-time applications. While its accuracy was satisfactory for larger objects, it struggled with smaller ones. This suggests a need for improvement in YOLOv5's ability to localize and identify a wider range of objects within complex driving environments, such as those on Indian roads.

Siddhant

We compared the performance of Faster RCNN models and YOLOv5 on UKNEC dataset and the models were trainde on IDD and KITTI 2d dataset respectively and only the common classes were use for analysis. We found that Faster R-CNN was more accurate, especially on the KITTI dataset. Faster R-CNN was model proved to be effective in accurately detecting objects of various size and shape. Despite its slower processing time in senarios where precision is critical and which may affect the application the Fast RCNNs high accuracy makes it valuable for such scenarios TIDE tool was used for identifying specific errors, pointing to the need for improvements in bounding box precision and localization accuracy.

Aishiki Bhatacharya

Our study's main purpose was to do a comparative analysis of two object detection models using the Indian Driving Dataset to see which model performs better under different conditions. We concluded that YOLOv5 was detected very quickly. This is beneficial for real-time processing, but the accuracy could be low in cluttered environments. On the other hand, Faster R-CNN was slower but had higher accuracy and was more reliable for detecting many different objects. This suggests a trade-off between speed and accuracy that must be considered when selecting a model for specific applications in autonomous driving and related fields. The TIDE framework analysis helped us determine areas where the models can improve and enhance the overall performance.

References

[1] BOLYA, D., FOLEY, S., HAYS, J., AND HOFFMAN, J. Tide: A general toolbox for identifying object detection errors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (2020), Springer, pp. 558–573. 3

[2] LIU, H., SUN, F., GU, J., AND DENG, L. Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode. *Sensors 22*, 15 (2022), 5817. 1

[3] REN, S., HE, K., GIRSHICK, R., AND SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems 28* (2015). 2

[4] VARMA, G., SUBRAMANIAN, A., NAMBOODIRI, A., CHANDRAKER, M., AND JAWAHAR, C. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE winter conference on applications of computer vision (WACV)* (2019), IEEE, pp. 1743–1751. 2