Final Report

Team Members: Jashwanth Neeli, Megha Kaavali Mahadevappa

Object Detection for Autonomous Vehicles

Abstract:

The detection and tracking of objects in computer vision systems is a critical and challenging field with extensive applications across various industries, including surveillance, autonomous robot navigation, and vehicle navigation. The increased accessibility of computing power and extensive public datasets has driven advancements in this area. Object tracking algorithms aim to segment the region of interest in an image, monitor its movement and location, and identify any significant blobs. Effective tracking requires an object detection mechanism, either in each frame or whenever an object first appears in the video sequence. The availability of high-performance processors and affordable, high-quality video cameras has increased the demand for automated object tracking in video sequences. Recently, the focus of computer vision research has shifted towards multiple object identification and tracking in dynamic environments. This paper discusses various object detection and tracking algorithms.

Introduction:

The field of computer vision relies heavily on object detection, which involves identifying and locating objects within an image. This technology not only recognizes the types of objects present in an image but also precisely pinpoints their positions using bounding boxes. Each detection is assigned a confidence score, which indicates the system's level of certainty in its accuracy. This capability is crucial in various applications, including autonomous driving and security systems, as it allows machines to effectively interpret and interact with the visual world.

Object detection model training teaches neural networks to recognize and locate objects using labeled images. The model learns to identify object features and patterns, using this knowledge to make predictions on new images for real-world use.

The YOLO (You Only Look Once) series is known for its efficient and speedy object detection architectures. YOLOv8, the latest in the series, introduced by Ultralytics, offers significant enhancements in architecture and usability, delivering top-tier performance on standard benchmarks like the COCO dataset and specialized tests such as Robo Flow 100.

Dataset:

"Our main goal is to utilize a customized dataset, specifically a video streaming dataset."

The input video file contains footage of the streets of Chicago, which we recorded. It includes various types of vehicles such as buses, bikes, cars, and bicycles, as well as streetlights and pedestrians. We utilized a COCO dataset with 80 labels to help identify these objects.

The input data file has been converted into 3500 images with corresponding labels. These images have been further classified into training and validation sets. The training set consists of approximately 2850 images, while the validation set contains around 650 images.

Literature Review:

Object Detection, classification, and object tracking are the three primary phases in image processing.

Object Detection Methods:

Background Subtraction for Object Detection

Background subtraction is a fundamental technique in the realm of object detection, particularly effective in static camera setups. This method involves analysing the differences in pixel values between the current frame and a reference background model. The reference model is typically generated from a series of previous frames to represent the scene without any moving objects. By subtracting the current frame from this background model, regions corresponding to moving objects can be identified. This technique is highly efficient in scenarios with a relatively static background, making it popular in surveillance systems, traffic monitoring, and other applications where the scene's background remains largely unchanged over time.

Optical Flow in Object Detection

Optical flow techniques are another cornerstone in object detection, focusing on the motion patterns within a sequence of frames. Optical flow calculates the apparent motion of objects between consecutive frames by analysing the displacement of image features. This information is then used to identify and track moving objects within the scene. By clustering motion vectors, optical flow can effectively detect objects even in complex and dynamic environments. This method is particularly useful in applications where precise motion tracking is crucial, such as autonomous driving, robotics, and video analytics.

Complementary Methods for Object Detection

Enhance the accuracy and robustness of object detection, it is common to employ a combination of methods, such as frame differencing, optical flow, and background subtraction. Frame differencing involves comparing the differences between consecutive frames to detect motion, which can quickly identify changes in the scene. Optical flow provides detailed motion information, allowing for the detection of moving objects even when the background is not static. Background subtraction excels in static environments, efficiently isolating foreground objects. By integrating these complementary techniques, object detection systems can achieve higher accuracy and reliability, adapting to various scenarios and overcoming the limitations of individual methods.

Object Classification Methods:

Motion-Based Object Classification

Researchers, like Hitesh A. Patel, have developed motion-based classification techniques that do not rely on predefined patterns or models. Instead, these methods classify objects based on their movement characteristics captured over time. This approach is advantageous in dynamic environments, where object behaviour can vary widely. However, a significant challenge of motion-based classification is its difficulty in identifying stationary objects. Objects that remain still or move very slowly can be overlooked or misclassified since the method relies on detecting motion. Integrating additional techniques is necessary to ensure comprehensive object detection and classification, especially in surveillance and monitoring systems where both moving and stationary objects are of interest.

Texture-Based Object Classification

Texture-based image classification uses texture information to differentiate between objects. It analyses intensity patterns within an image to identify objects based on surface characteristics. While computationally intensive, this method is effective in scenarios where surface detail is crucial, such as medical imaging and material inspection.

Shape-Based Object Classification

Researchers like Salve and Jondhale focus on shape-based classification methods, which use simple pattern-matching algorithms to compare the shapes of detected objects against predefined templates. These methods are suitable for graph-matching applications and can effectively classify objects with distinct shapes. However, they may be less effective in complex scenes with overlapping objects or varying orientations. Despite this, shape-based classification remains valuable in fields such as robotics for accurate shape recognition.

Object Tracking Methods:

Overview of Point Tracking Technique

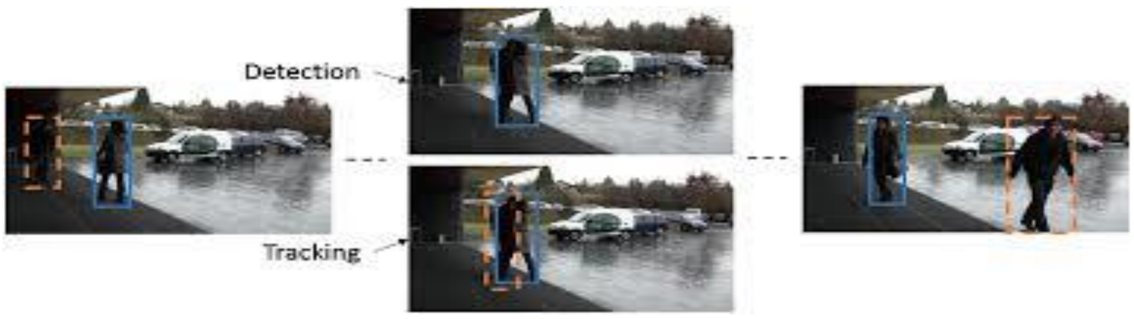
Point tracking is a method used to track objects by focusing on specific features within a frame. These features are tracked across frames to determine object movement. Despite its simplicity and efficiency for real-time applications, point tracking faces challenges such as erroneous detection and occlusion. Erroneous detection happens when background elements are mistakenly identified as part of the object, leading to inaccurate tracking. Occlusion occurs when the object is temporarily obscured, causing the tracker to lose track of focal points. Advanced algorithms aim to predict and recover from occlusions to improve tracking robustness.

Analysis of Kernel Tracking Implementation

Kernel tracking uses a template called a kernel to represent and track objects between frames, providing a more comprehensive representation of an object's motion than simple point tracking. This method is effective in dynamic environments and scenarios where object shape and size may vary, such as sports tracking and gesture recognition.

Significance of Silhouette Tracking

Silhouette tracking uses object models to identify and track regions of interest within images by creating a silhouette or outline of the object to be tracked. It is highly effective in surveillance applications, handling complex object shapes and maintaining high tracking accuracy in cluttered and dynamic environments.



Methodology:

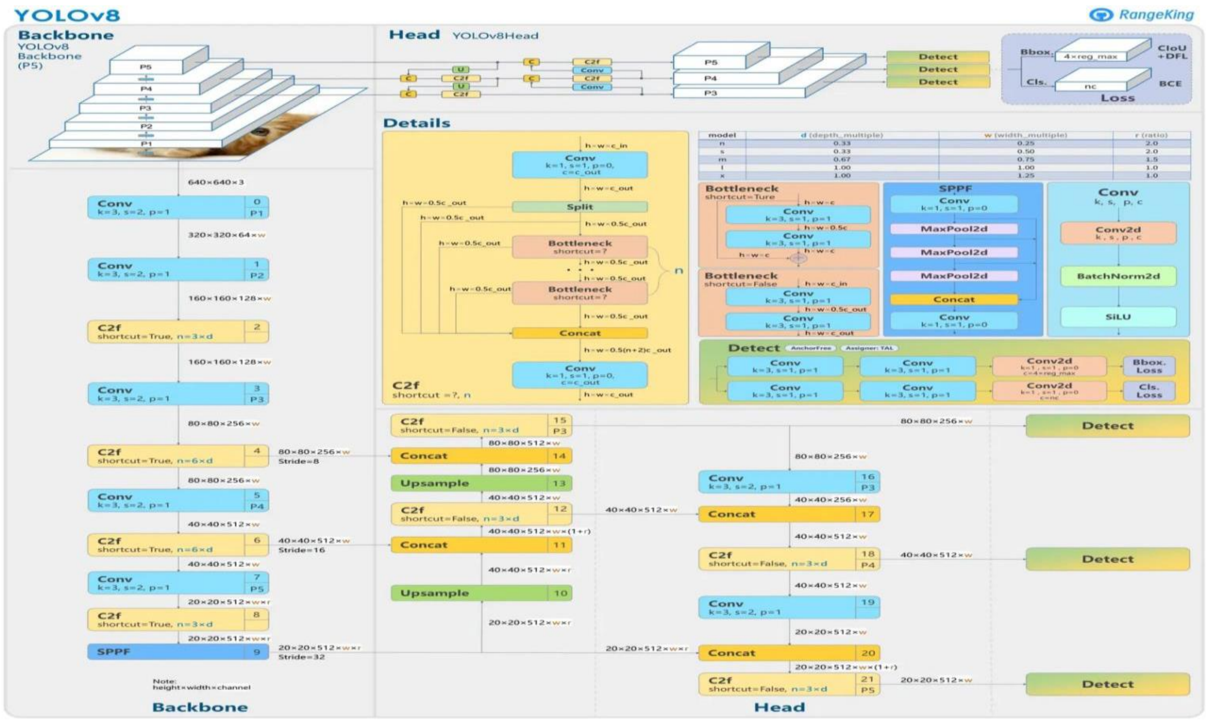
Performance of YOLOv8 on COCO and Roboflow 100

YOLOv8, the latest iteration of the "You Only Look Once" (YOLO) series of object detection models, showcases remarkable accuracy when evaluated on the COCO (Common Objects in Context) dataset. The medium-sized model in the YOLOv8 family, known as YOLOv8m, achieves an impressive mean Average Precision (mAP) of 50.2% on the COCO dataset. This metric underscores the model's ability to accurately detect and classify objects across a diverse range of categories and challenging scenarios presented in COCO.

In addition to its performance on COCO, YOLOv8 was tested against the Roboflow 100 dataset, a comprehensive dataset specifically designed to evaluate model performance across various task-specific domains. This dataset includes a wide array of images and annotations, reflecting different real-world conditions and tasks. When benchmarked against this dataset, YOLOv8 demonstrated a significant improvement over its predecessor, YOLOv5. The substantial margin by which YOLOv8 outperformed YOLOv5 highlights the advancements made in this latest model, particularly in terms of accuracy, efficiency, and robustness across different tasks and environments.

The detailed performance analysis of YOLOv8, including comparisons with YOLOv5 and other models, is discussed in the later sections of the article. These sections delve into the specific improvements and innovations introduced in YOLOv8, such as enhanced network architecture, better training techniques, and more efficient use of computational resources, which collectively contribute to its superior performance.

By integrating advanced features and leveraging state-of-the-art techniques, YOLOv8 sets a new benchmark in the field of object detection, making it a valuable tool for various applications ranging from autonomous driving to real-time video surveillance and beyond.



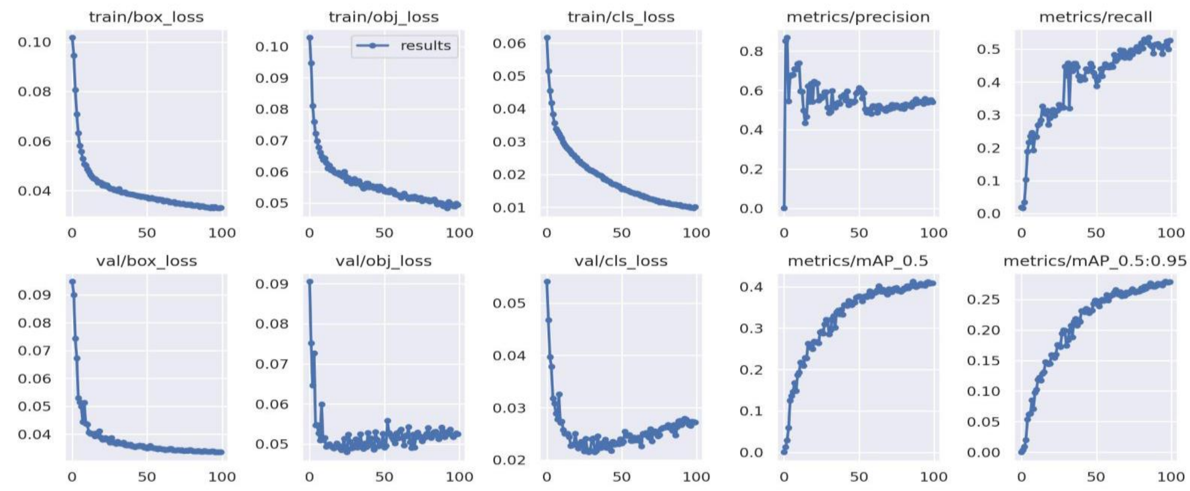
In object detection, the primary objective is to identify and locate multiple objects within an image. This task differs from image classification, where the focus is solely on categorizing the entire image into a single class. In object detection, the model aims to accurately predict all the objects present in the image, regardless of whether they belong to the same or different classes.

Object detection models achieve this by dividing the prediction process into two distinct parts. First, they predict the location of each object by generating a bounding box through a regression process. This bounding box defines the coordinates that encapsulate the object within the image. Next, the models assign a class label to each predicted bounding box using classification techniques. This two-step approach ensures that the model not only identifies where objects are located but also determines what these objects are.

One of the advancements in object detection is the development of anchor-free detection methods. Traditional anchor-based methods involve generating numerous predefined bounding boxes (anchors) at different scales and aspect ratios, which can lead to many box predictions. Anchor-free detection, on the other hand, reduces the number of box predictions by directly predicting the center points and sizes of objects. This reduction in predictions speeds up the Non-Maximum Suppression (NMS) process, a critical post-processing step in object detection.

NMS is used to filter out redundant and overlapping bounding boxes by selecting the most confident predictions while suppressing the less likely ones. Since NMS is a computationally intensive process, reducing the number of candidate detections through anchor-free methods significantly enhances the overall efficiency and speed of object detection models. This improvement is particularly beneficial for real-time applications where quick and accurate detection is crucial.

Results:

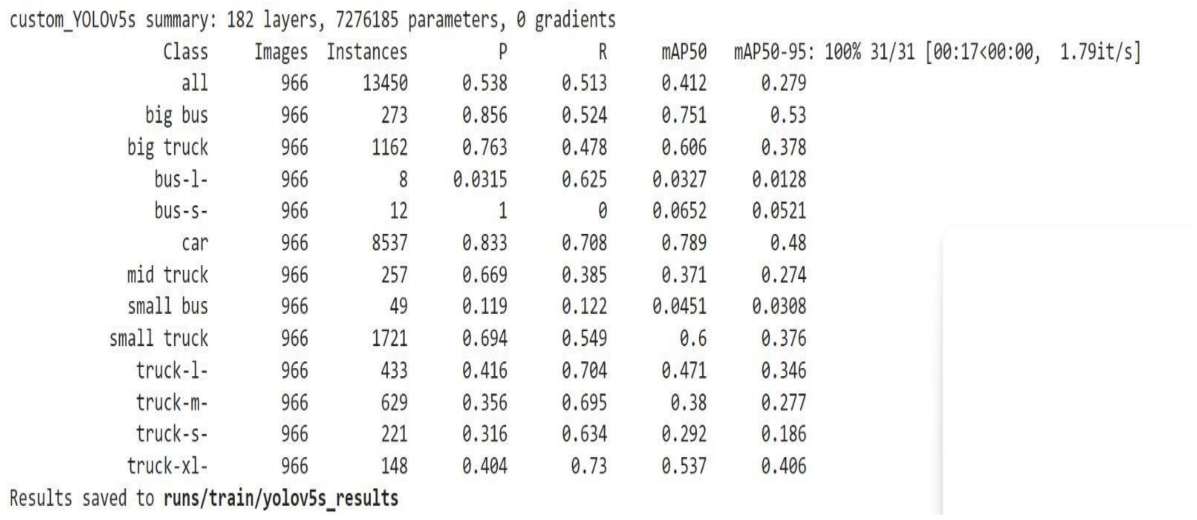


The model has demonstrated significant improvement across all loss metrics, including box loss, objectless loss, and classification loss, indicating its effective learning from the training data. This improvement is reflected in both precision and recall, which have increased over time, highlighting the model's enhanced ability to accurately identify and classify objects.

Moreover, there has been a notable increase in mean Average Precision (mAP) scores at Intersection over Union (IoU) thresholds of 0.5 and in the range from 0.5 to 0.95. These improvements suggest that the model's predictions are becoming more closely aligned with the ground truth annotations in terms of both object presence and precise location, demonstrating a higher overall accuracy and reliability in object detection tasks.

Observations on Class Performance

It has been observed that certain classes, such as "bus-l-" and "bus-s-", exhibit very low-performance metrics. This underperformance can likely be attributed to a few key factors. One possible reason is the limited number of instances for these classes in the dataset, which restricts the model's ability to learn and generalize effectively. Additionally, the high similarity of these classes with other vehicle types could lead to frequent misclassification, as the model struggles to distinguish between visually similar categories.



```
custom_YOLOv5s summary: 182 layers, 7276185 parameters, 0 gradients
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 31/31 [00:17<00:00,  1.79it/s]
                   all        966      13450      0.538      0.513      0.412      0.279
               big bus        966        273      0.856      0.524      0.751       0.53
             big truck        966       1162      0.763      0.478      0.606      0.378
                bus-l-        966          8     0.0315      0.625     0.0327     0.0128
                bus-s-        966         12          1          0     0.0652     0.0521
                   car        966       8537      0.833      0.708      0.789       0.48
             mid truck        966        257      0.669      0.385      0.371      0.274
             small bus        966         49      0.119      0.122     0.0451     0.0308
           small truck        966       1721      0.694      0.549        0.6      0.376
              truck-l-        966        433      0.416      0.704      0.471      0.346
              truck-m-        966        629      0.356      0.695       0.38      0.277
              truck-s-        966        221      0.316      0.634      0.292      0.186
             truck-xl-        966        148      0.404       0.73      0.537      0.406
Results saved to runs/train/yolov5s_results
```

The noticeable variance in performance across different vehicle types highlights a significant area for potential improvement. To enhance the model's accuracy, several strategies could be considered:

1. **Balanced Data Distribution**: Ensuring a more balanced dataset with enough instances for each class can help the model learn more effectively. This can be achieved through techniques such as data augmentation, synthetic data generation, or targeted data collection efforts to increase the representation of underperforming classes.
2. **Enhanced Feature Extraction**: Developing more robust feature extraction methods tailored specifically to the challenging classes can improve the model's ability to differentiate between similar objects. This may involve utilizing advanced neural network architectures, implementing specialized preprocessing techniques, or incorporating domain-specific knowledge to enhance feature discrimination.
3. **Extended Training Epochs**: Increasing the number of training epochs can provide the model with additional opportunities to refine its understanding of the dataset. This extended training period allows the model to better capture the nuances and variations within each class, potentially leading to improved performance across all categories.

Conclusion:

Object detection models are trained using images that are precisely labeled with the locations of various objects. This labeling process enables the models to accurately learn how to identify and categorize different objects.

However, the performance of these models can vary significantly across different object classes.

To address this variability, it is crucial to optimize the models continually. This can be achieved by analyzing instances of misclassification, augmenting the training data for underperforming categories, and further tuning the model parameters. These steps help refine the model's accuracy and robustness across all object classes.

Future Work:

Implementation of lane line detection.

Training the model, and checking the recall, precision, and map score for lane line detection.