Object Detection in Automotive Domain: A Supervised Learning Comparison

Kevin Lin, Jiamu Li and Ashwin Sharan

Abstract

This scientific paper proposes the development and comparison of object detection models for detecting cars, pedestrians, and street signs in the automotive domain. The study aims to train and assess the performance of each model using supervised learning approaches, including YOLO, faster R-CNN, and SVM models. The gathered information will be tagged with bounding boxes or masks to locate the objects in the image. The primary objective of the project is to efficiently detect and classify objects using various supervised learning approaches to address the problem of object detection.

Introduction

The problem of object detection in the automotive domain is critical for the development of autonomous vehicles. The effectiveness of an autonomous vehicle is heavily reliant on its ability to detect and localize objects in real-time driving scenarios. Therefore, the development of accurate and efficient object detection models is crucial for the advancement of this technology. In this study, the authors propose to compare the performance of different supervised learning approaches, such as YOLO, Faster R-CNN, and SVM models, for detecting cars, pedestrians, and street signs in the automotive domain. The objective of the project is to develop an effective object detection model that can be deployed on a real-world autonomous vehicle. The proposed project will include data collection, preparation, object detection model development, and evaluation. The best performing model will be selected based on the evaluation results to demonstrate its effectiveness in detecting and localizing objects in real-time driving scenarios.

Related work

The field of object detection has seen various approaches to address the challenges it poses. One such approach is the probabilistic method proposed by Feng et al [3], which utilizes Bayesian neural networks (BNN). In this study, the authors have used two common sources for acquiring data, namely COCO [1] and KITTI [6].

Zhao et al. [4] have also proposed a similar approach to object detection, where they have used a combination of tools such as SVM classifiers with HOG feature extraction, YOLO, and Faster R-CNN. Their study focused specifically on pedestrian and face detection. They have also used some of the same evaluation methods as in this paper.

In another study by Chen and Elangovan [2], the authors have employed Faster R-CNN for object sorting, which also employs some of the same evaluation methods as used in this study. However, their study has focused more on robotic and manufacturing applications.

Background

Models used in supervised learning approach for object detection:

YOLO:

You Only Look Once (YOLO) is a popular object detection algorithm that was introduced in 2016 by Joseph Redmon et al. It is a convolutional neural network (CNN) architecture that performs object detection in real-time. YOLO is unique in that it uses a single neural network to predict the bounding boxes and class probabilities for objects in an image. This is in contrast to other object detection algorithms that use a two-stage approach, where regions of interest are first identified and then classified.

The YOLO algorithm divides the input image into a grid of cells and for each cell, predicts a set of bounding boxes and class probabilities for each object. The bounding boxes are represented as four values: the center coordinates, width, and height of the bounding box. The class probabilities represent the likelihood of each object belonging to a particular class. The YOLO algorithm is trained on a large dataset of annotated images using a loss function that penalizes incorrect predictions of bounding boxes and class probabilities.

One of the key features of YOLO is its speed. YOLO can detect objects in real-time on a standard CPU, achieving frame rates of up to 45 frames per second.

Faster R-CNN:

Faster R-CNN is a popular two-stage object detection algorithm that was introduced in 2015 by Shaoqing Ren et al. It is a convolutional neural network (CNN) architecture that performs object detection by first proposing regions of interest (ROIs) in an image and then classifying the objects within these regions. The RoIs are generated using a Region Proposal Network (RPN), which is a small CNN that is trained to predict objectness scores and bounding box offsets for each anchor box in an image.

The RPN generates a set of RoIs that are then passed to a second CNN for classification and bounding box regression. The second CNN takes the proposed RoIs as input and generates class probabilities and refined bounding box coordinates for each object. The refined bounding box coordinates are obtained by predicting offsets from the proposed RoIs to the ground truth bounding boxes.

Faster R-CNN is trained end-to-end using a joint loss function that penalizes incorrect predictions of objectness scores, class probabilities, and bounding box offsets. The algorithm is trained on a large dataset of annotated images, and the weights of the CNNs are learned using backpropagation.

One of the advantages of Faster R-CNN is its high accuracy, which is achieved by using a two-stage approach that separates region proposal from classification. However, this comes at the cost of speed, as the algorithm requires two CNNs to be run in sequence.

SVM:

Support Vector Machines (SVM) are a type of supervised learning algorithm used for classification and regression analysis. They have been widely used in the field of object detection. SVMs use a kernel function to map input data into a higher dimensional feature space, where the data can be separated by a hyperplane. In the context of object detection, the features used by the SVM classifier are typically extracted from the input image using feature extraction techniques.

One such feature extraction technique is the Histogram of Oriented Gradients (HOG), which is a widely used method for detecting object boundaries in images. The HOG method works by calculating the gradient magnitude and orientation of the image pixels and then grouping them into small cells. The gradients within each cell are then normalized using a block normalization method to produce the final feature vector.

The HOG features are then used as input to an SVM classifier, which learns to distinguish between different objects in the image. During training, the SVM optimizes the hyperplane that best separates the positive and negative examples in the feature space. The resulting SVM model can then be used for object detection by extracting HOG features from a new image and classifying them using the trained SVM.

Using SVMs with HOG features has shown promising results for object detection in various applications, including pedestrian detection, vehicle detection, and face detection.

Methodology

The following steps are taken to train YOLOv8 and ResNet-50-FPN models

1. Data Collection: The COCO 2017 Train dataset's car, person, and stop sign classes [1] are combined with the INRIA person dataset [5] to create a custom dataset for the YOLO and Faster R-CNN models. This custom dataset contains 7470 images in total, with 9611 instances of vehicles, 11131 instances of persons, and 1983 instances of stop signs.

2. Data Conversion: Annotations are converted to Darknet TXT format for the YOLOv8 model and VOC XML format for the Faster R-CNN ResNet-50-FPN model. This step ensures that the data is in a format that is compatible with the respective models.

3. Data Shuffling: The data is shuffled to increase the accuracy of the models. This step is essential to avoid any bias in the data and ensure that the models can generalize well.

4. Data Splitting: The shuffled data is then split into three parts: training, validation, and testing. 80% of the data is selected for training, 10% for validation, and 10% for testing. This split ensures that the models are trained on a significant amount of data, and their performance is evaluated on previously unseen data.

5. Data Preprocessing: Before training the models, the images are normalized and resized to 640x640 in the models' configuration phase. The bounding box values are also modified accordingly to ensure that the objects' locations are accurately represented in the models.

6. Model Training: The YOLOv8 and Faster R-CNN ResNet-50-FPN models are trained on the custom dataset using the training set. The models are trained to detect and classify vehicles, persons, and stop signs in the images.

7. Model Validation: The validation set is used to evaluate the models' performance and fine-tune their hyperparameters to improve their accuracy.

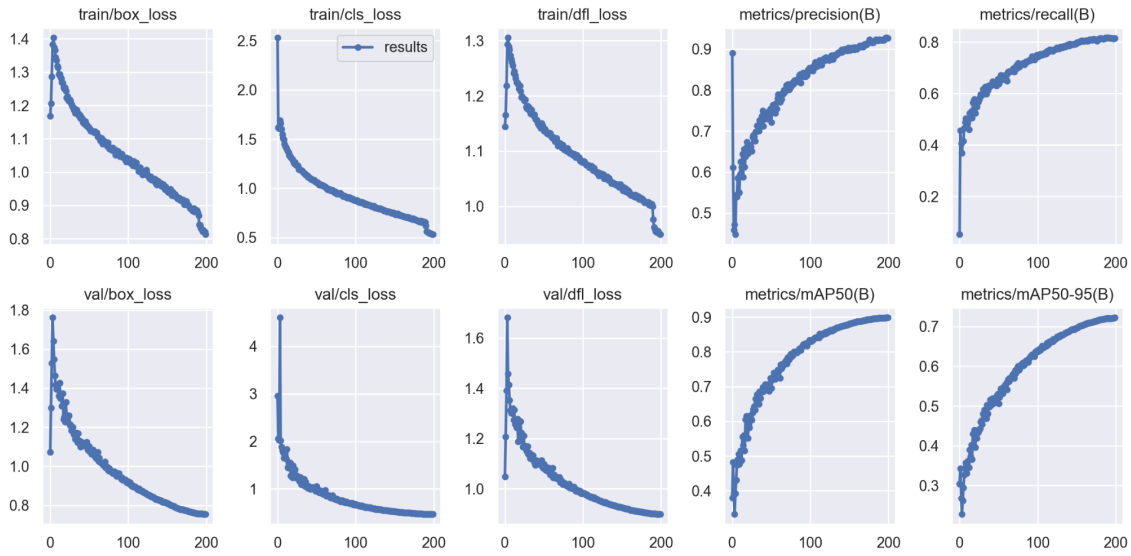

Figure 1: Yolo training and validation.

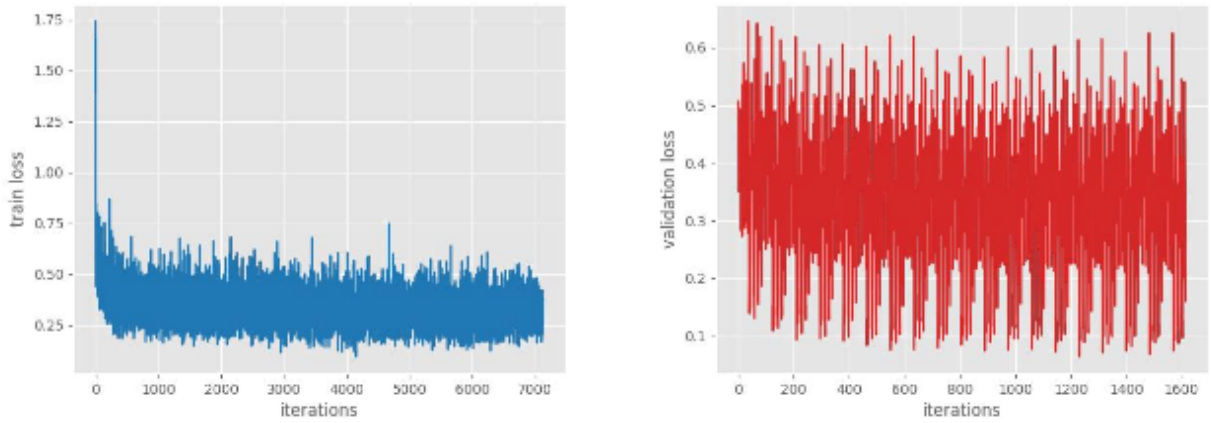

Figure 2: RCNN training and validation loss.

8. Model Testing: The testing set is used to evaluate the models' final performance and assess their ability to generalize to previously unseen data.

9. Visual Evaluation: The models are run on pre recorded video of street traffic as shown in the figures below.

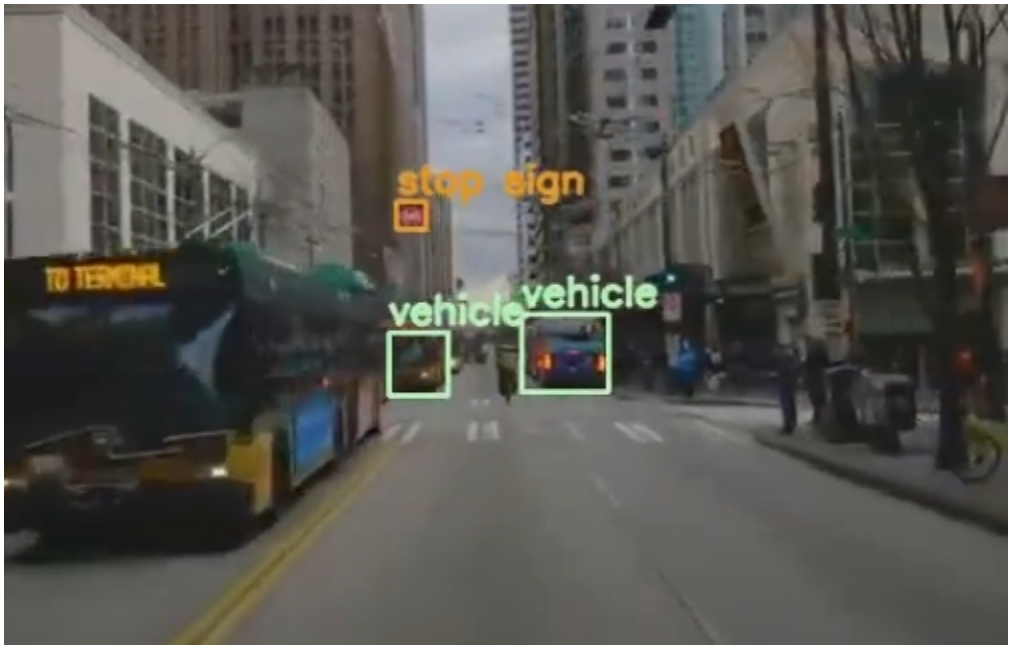

Figure 3: Resnet50 visualization.

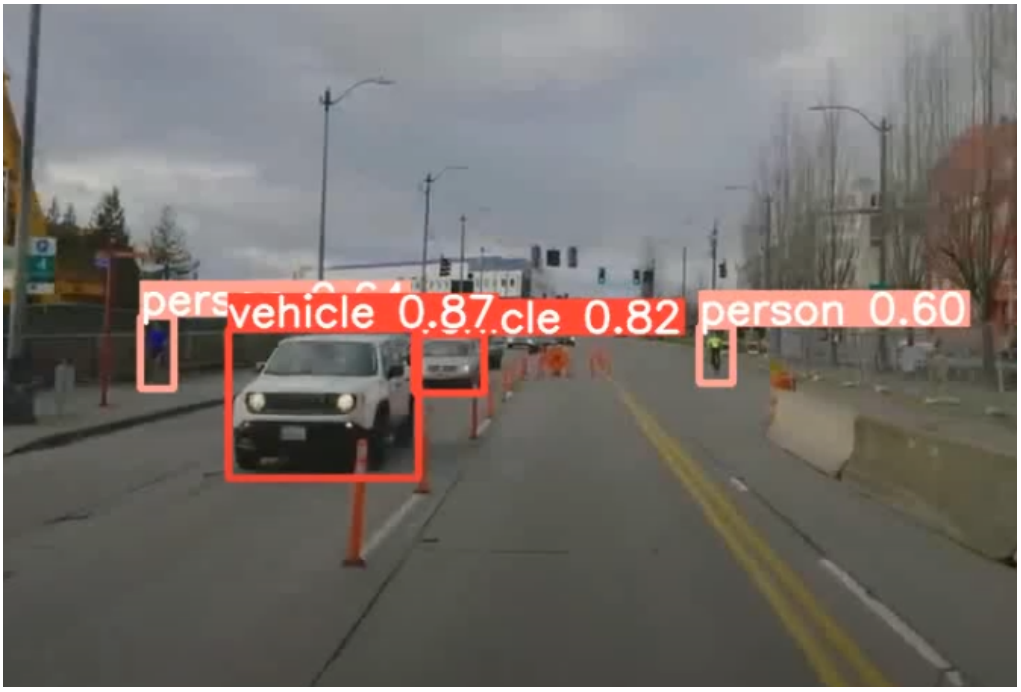

Figure 4: YOLO visualization

The following are the steps taken to train the SVM model.

1. Data collection: The COCO 2017 Train dataset's person and stop sign classes[1] are cropped according to their bounding box annotation size and resized to 64 x 64. Also detailed car images

were taken from the KITTI vision benchmark suite [6]. The dataset contains 8792 images of cars, 11237 images of people, 1983 images of stop-sign and 16171 non-domain images.

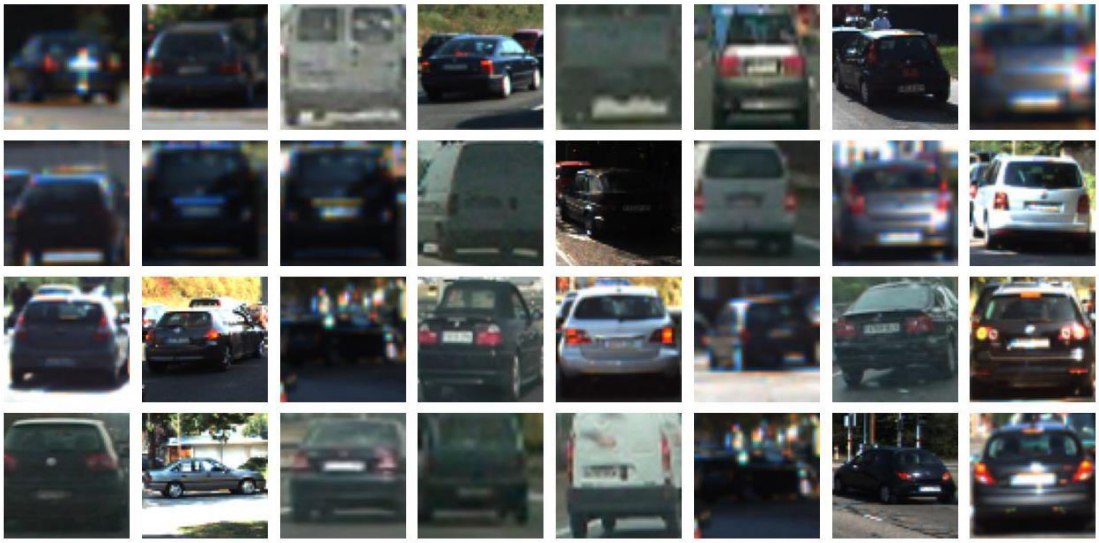

Figure 5: Car Samples

Figure 6: Non-Domain Samples

2. Feature Extraction: The Histogram of Oriented Gradients are extracted for each individual domain and the non domain Image as seen in the visualizations below. To classify more accurately, histogram and binned color features are also extracted.

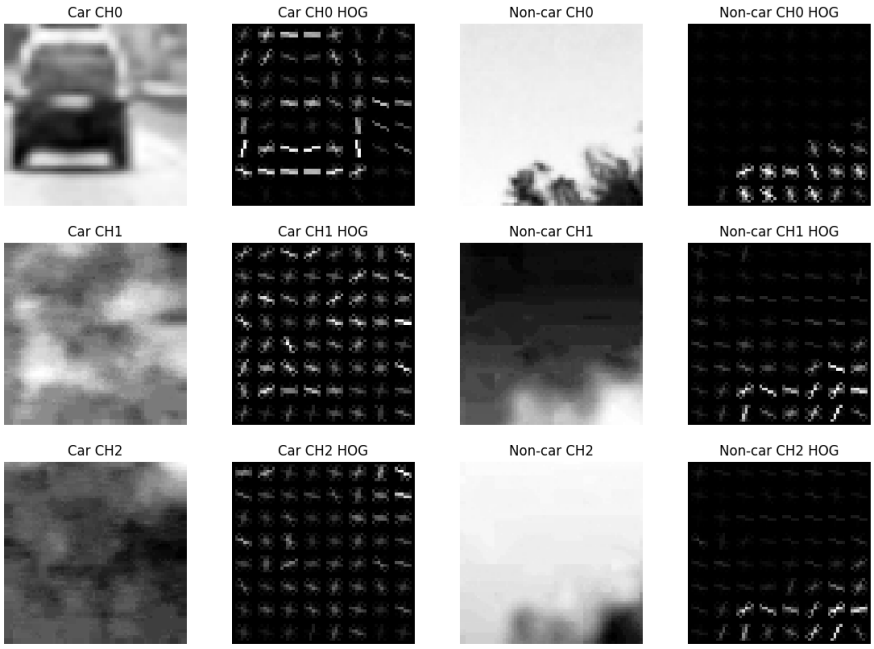

Figure 7: HOG visualization for cars.

3. Data preparation: Each domain image feature is stacked with non-domain features given an output label of 1 for domain and 0 for non-domain. Then split into 80% for training and 20% for testing.

4. Classification: Linear SVC to train 3 separate models on the data to classify cars, person and stop sign in images.

5. Sliding window: HOG window search routine is used, where it scans select portions of the image and checks for presence of a domain object.

6. Temporal heat mapping: Heat mapping of successful hits in the frame to reduce false positives and improve accuracy.

7. Visual Evaluation: The models are run on pre recorded video of street traffic as shown in the figures below.

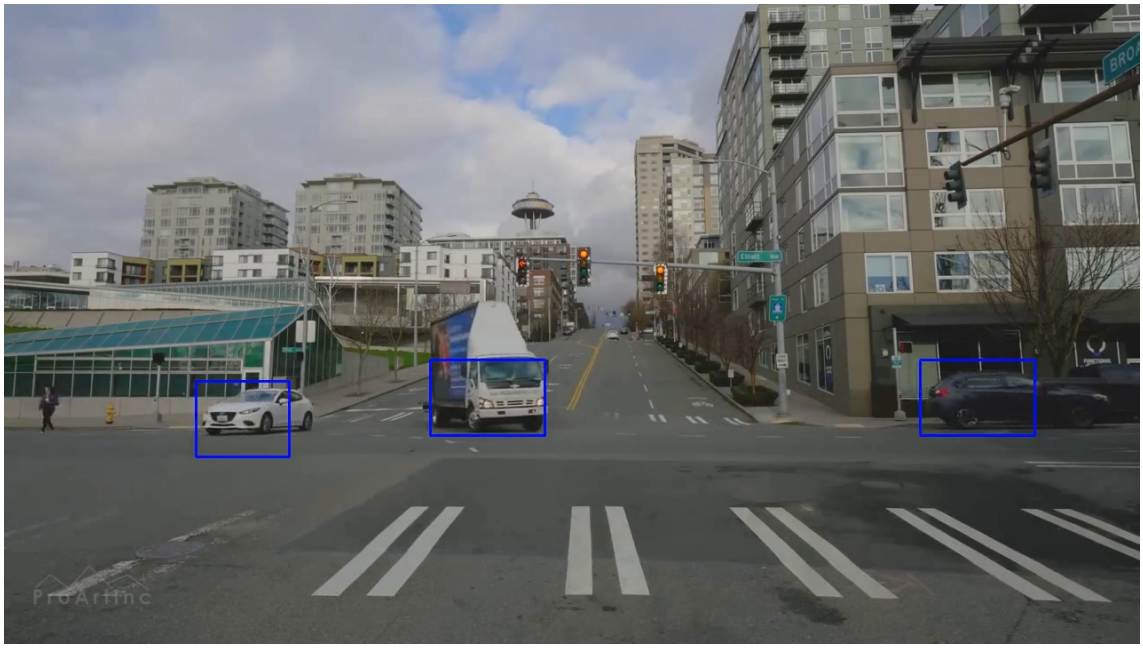

Figure 8: SVM visualization.

Evaluation

SVM:

In the scope of this study, the SVM model was found to be insufficient in detecting pedestrians and stop signs. However, it performed well in detecting vehicles due to the availability of a large number of diverse images from the KITTI dataset [6] with various angles and better compositions. Additionally, a vast collection of non-domain images of empty roads enabled the model to differentiate between the background and vehicles effectively.

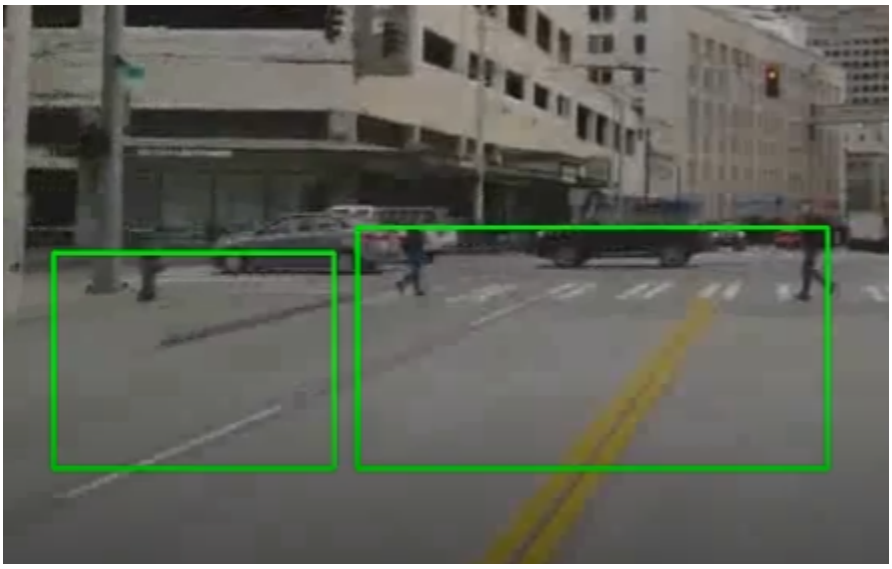

Figure 9: Pedestrian visualization for SVM.

Several approaches were attempted to improve pedestrian and stop sign detection, such as image resizing, increasing the number of samples for domain and non-domain images, and varying the hyperparameters for HOG feature extraction. However, these methods were unsuccessful due to the high variability in pedestrian appearance, which includes changes in clothing, body posture, and lighting conditions. A robust model for pedestrian detection requires the ability to handle complex variations in appearance and adapt to changes in the environment. SVM is limited in this aspect as a linear classifier and may not be able to capture the nonlinear relationships between image features and pedestrian detection accurately. Additionally, stop signs were harder to detect due to their small size and the vast non-domain space in video streams.

To improve future research, obtaining more images of pedestrians in traffic-rich areas and increasing the collection of empty sidewalks and traffic ways will enhance the region of interest. Moreover, further research could investigate adjusting hyperparameters for HOG feature extraction to improve pedestrian and stop sign detection.

YOLO:
The model was found to be suitable in accurately detecting vehicles, pedestrians and stop signs with a speed of 45fps in real-time video.

In terms of accuracy:

Domain	Box Precision	Bounding Box Recall	Mean Average Precision (MAP)
All	0.927	0.815	0.899
Vehicles	0.893	0.703	0.829
Person	0.929	0.826	0.907
Stop sign	0.959	0.91	0.96

Faster R-CNN:
The model was found to be suitable in accurately detecting vehicles, pedestrians and stop signs with a speed of 15fps in real-time video.

In terms of accuracy:

Domain	Box Precision	Box Recall
All	0.89	0.804
Vehicles	0.946	0.783
Person	0.779	0.746
Stop sign	0.946	0.8844

Conclusion

In conclusion, this scientific paper presented the development and comparison of object detection models for detecting cars, pedestrians, and street signs in the automotive domain. Three supervised learning approaches were implemented: YOLO, faster R-CNN, and SVM models. The models were trained and evaluated using the Coco [1] and KITTI [6] datasets, with the gathered information tagged with bounding boxes or masks to locate the objects in the image.

Our experimental results showed that YOLO outperformed the other models with a 0.90 mean average precision (MAP) and a speed of 40 frames per second (fps). Faster R-CNN and SVM models showed lower performance in terms of both accuracy and speed. Therefore, we recommend the use of YOLO for object detection tasks in the automotive domain.

References:

[1] Tsung-Yi Lin, Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., … Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. CoRR, abs/1405.0312. Retrieved from http://arxiv.org/abs/1405.0312

[2] Chen, Pengchang, and Vinayak Elangovan. “OBJECT SORTING USING FASTER R-CNN.”

https://arxiv.org/ftp/arxiv/papers/2012/2012.14840.pdf. Accessed 2012.

[3] Feng, DI, et al. “A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving.” *arXiv*, 20 November 2020, https://arxiv.org/abs/2011.10671. Accessed 27 April 2023.

[4] Zhao, Zhong-Qiu, et al. “Object Detection with Deep Learning: A Review.” *arxiv*, https://arxiv.org/pdf/1807.05511.pdf. Accessed 16 4 2019.

[5] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat and Pierre Alliez. “Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark”. IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2017.

[6] Andes Gejger, Philip Lenz, Raquel Urtasun, Are we ready for Autonomous Driving? “The KITTI Vision Benchmark Suite”. Conference on Computer Vision and Pattern Recognition (CVPR). 2012.