International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)

IJCI

المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)

October 2025

Explainable AI: Objection Detection in Automated Vehicles using YOLO

Osama Mohammed Dad
College of Computer Science and Engineering, Taibah University, Madinah, Saudi Arabia

Abdul Ahad Siddiqi
Associate Professor, Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah, Saudi Arabia
asadqi@taibahu.edu.sa

Abstract

There is a recognized need for explainable artificial intelligence (XAI) in today's age as AIs are getting integrated more and more into our lives, from everyday tasks to huge fields where AI decisions are in a critical environment and could impact people's lives, XAI aims to make us understand the model behavior and decisions using developed methods, one such field is autonomous vehicles where there are many tasks to that uses different models, object detection in one of these tasks where the vehicles need to observe and assess the environment around them, several studies have shown the importance of XAI in the fields autonomous vehicles, and explained their actions and detections. However, true explainability has not yet been reached, with the complexity of Deep Neural Networks (DNN) and the tasks of AV, and the impact of performance and explainability motivates more studies of explainability. In this study, we train YOLOv11 and YOLOv12 models on the KITTI and a cyclist dataset and implement the explainability techniques Grad-CAM++ and Eigen-CAM to explain the decision behind them. The models showcased great performance, achieving 0.93 precision, 0.93 recall, 0.95 mAP@50, and 0.83 mAP@50-95 for YOLOv11 and similar results from YOLOv12. The heatmaps generated from Grad-cam and

Eigen-cam showcased where the model focuses on for the detections, overall explaining that this model will increase our trust and safety of AVs.

Keywords: Explainable AI, XAI, YOLO, KITTI, Grad-CAM, Eigen-CAM, Computer Vision, Autonomous Vehicles.

1. Introduction

Artificial intelligence (AI) has made considerable advancements in every aspect of our lives. One of the many uses is in autonomous vehicles (AVs), from controlling the vehicle and decision-making to perceiving and planning its routes, making it fundamentally dependent on it for all its functions, and while it is establishing a basis for more secure and efficient transportation, as most road accidents are human error. It is concerning that the AI used in them is complex in processes and untransparent in how they work. It is a "black box" by nature, with decision-making that remains interpretable, even for specialists who made it. This deficiency in transparency presents significant obstacles, particularly in the realms of safety, trust, increasing public confidence, and following regulatory requirements. The incorporation of Explainable AI (XAI) within autonomous vehicles surfaces as a viable solution to mitigate these issues by augmenting the interpretability and transparency of AI systems, making users interpret the hows and whys of its conclusions while preserving their operational efficacy and increasing the trust, reliability, and ethics.

2. Background

2.1 Explainable AI (XAI):

AI is a cornerstone of technology today, with it being used in every sector of the world, from the government to everyday people, and with it the ability to learn, reason, and adapt; these characteristics give them the ability to process data and formulate conclusions making them effective in complex tasks, it is not surprising, while AI now is few decades old, the first couple AI were

interpretable, the introduction of Deep Neural Networks (DNN), with one of the first one to utilizes them was Alex K. with Alexnet/Imagenet in [1]. Which offers more efficient learning and parameter space but makes their decision opaque as the network consists of hundreds of layers and millions of parameters, which make them complex and can be considered a "black box," with Error! Reference source not found. showing earlier traditional model vs deep learning models in their complexity and explainability, and as these complex models are used in fields with critical decision making while giving unjustifiable, untrue, and not giving explanation for their decisions, the importance of explanations and transparency are increasing from its users [2].

To address the issues of opaque AI, the field of explainable artificial intelligence (XAI) was introduced [3], which proposes to develop AI that not only can provide an accurate prediction to maintain the effectiveness of current DNN level AI but also be explainable, enabling human users to understand and trust the AI decisions through an interpretable explanation for why the model made that decision or action and model transparency.

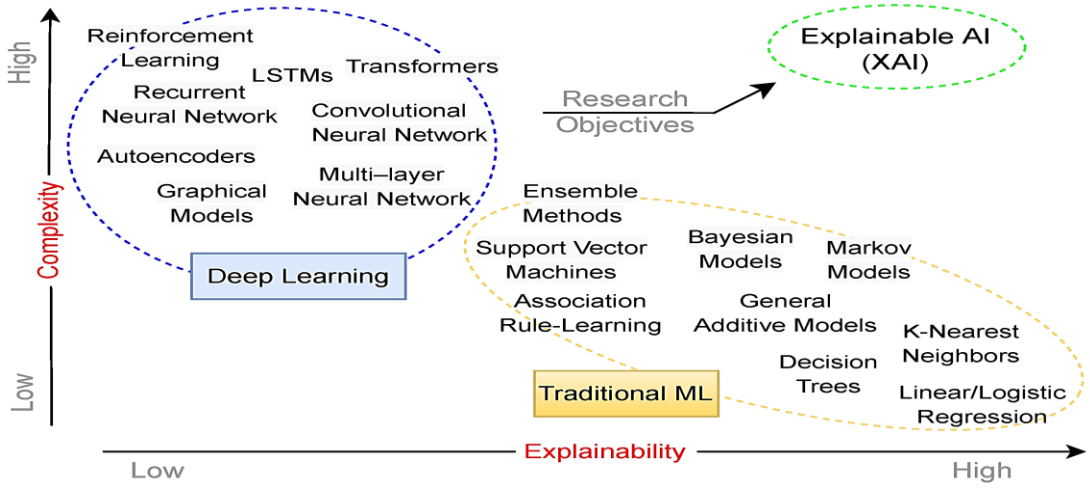


Figure (1): AI models in complexity and explainability [55]

2.2 Autonomous Vehicles:

Autonomous vehicles, also known as self-driving vehicles, are automobiles with advanced technologies, AI, and sensors that make them able to perceive, navigate, and operate in the real-world environment with no need for human intervention; AV systems are typically categorized into three tasks perception, planning, and control [4], with showing Error! Reference source not found. what a typical AV system look like, in this report we focused on perception tasks.

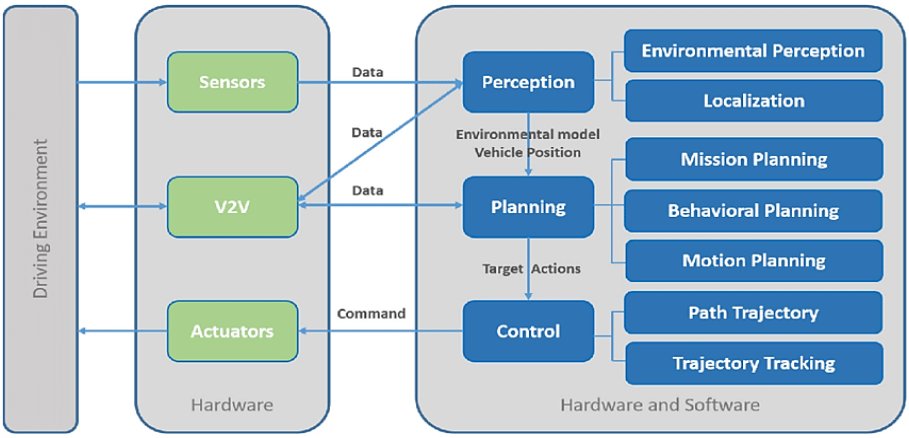


Figure (2): Typical autonomous vehicle system [4]

3. Problem Statement

Perception tasks are essential in AVs, where they can see and act upon the world. In object detection, the vehicle must detect people, other vehicles, signs, and traffic lights using deep learning models like region-based convolutional neural networks (faster R-CNN). You only look once (YOLO), where these models have high accuracy in detection, but they lack explainability and interpretability; this lack of transparency poses critical challenges, including safety risks, difficulty in debugging, reduced public trust, barriers to regulatory compliance, and in accidents it questions their decisions and compromises their safety in the real world.

4. Research Aim and Objectives

Research Aim: this research aims to develop and evaluate a framework that integrates XAI techniques into object detection in autonomous vehicles, addressing the challenges of transparency, interpretability, trust, and safety and investigating and evaluating previous XAI techniques and their application on object detection in autonomous vehicles, and creating a new framework for providing explainability for the AIs decision-making without compromising performance.

- Providing a state-of-the-art overview of XAI, existing techniques, and their limitations.
- Design, develop, and train a YOLO object detection model for autonomous vehicles.
- Design and develop a framework to explain the AI behind object detection systems in AVs.
- Evaluate the framework with existing techniques and how it improves on them.

5. Literature Review

This section presents an overview of previous literature on explainable AI in object detection for AV using different models, the XAI technique they applied, and their relevance for achieving interpretable models and safety and trust in autonomous vehicles.

Various existing reviews on XAI and AV provided valuable universal insight into the research challenges in understanding the concepts and structure of XAI and AV and its components and their integration; we classify the review papers into four categories: papers about standalone XAI, papers about AV alone with it components, how it works, papers about the object detectors, and papers that combine XAI in AV. We will start with the reviews of XAI, with the notable reviews being [2], [5], [6]. Providing an in-depth overview of XAI from concept, categories, and types of XAI

techniques, challenges, and the proposal of a responsible AI.

5.1 XAI Terms:

- **Understandability:** make the user understand the function and how the model works.
- **Post-hoc interpretability:** techniques to gain interpretability after the model training.
- **Transparency:** a model is intrinsically transparent when the user can understand, interpret, and predict the model's behavior by itself, and is divided into three categories:
  - Simulatable: the model can be simulated or thought about by humans.
  - Decomposable: the ability to explain each part of the model, such as input, parameter, and calculations, to understand, interpret, or explain how the model works without using additional tools, but that requires that every input be interpretable, which is not always possible.
  - Algorithmically transparent: understanding the process that produced the output from that input, while not understanding the deep architecture.[2]
- **Interpretability:** the model explaining how it works is, or itself is understandable.
- **Completeness:** How accurate were the explanations for explaining the model's inner workings?
- **Explainability:** the ability of the model to explain itself to the user so that they can understand the system as a whole, the connection between the input and the output, and how it got there. The explanation should answer at least one of these questions: [7]
  - What were the main factors in the decision?
  - Would changing a particular factor have changed the decision?

- Why did two similar-looking cases get different decisions and vice versa?

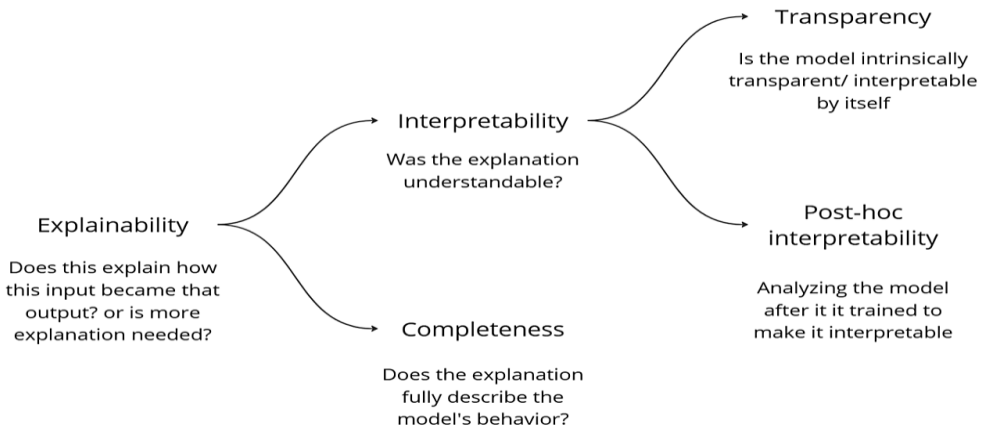


Figure (3): Taxonomy of XAI terms [7]

5.2 Techniques for XAI Implementation:

While there are old transparent models that are understandable by themselves, such as decision trees, linear/logistic regressions, and rule-based models, in this report, we focus on explaining the more complex "black box" models, such as Deep Neural Networks (DNN), which they cannot transparent. For that, we will explain to them after training them in the post-hoc stage, with Error! Reference source not found. summarizing the XAI techniques.

5.2.1 Based on Stage:

- **Ante-hoc**: consists of traditional AIs that are intrinsic or transparent and are models that are inseparable from themselves, also known as "glass box."
- **Post-hoc:** unlike ante-hoc, post-hoc is concerned with a complex "black box" model by techniques that we apply after the model training; examples of it are SHAP [8], LIME [9], and Grad-CAM [10].

5.2.2 Based on the Model: When implementing post-hoc techniques, we need to specify the method we will use.

- **Model-specific**: provide explainability for a specific model and use the model structure and algorithm to provide it.
- **Model-agnostic:** can be implemented universally in different types of models.

5.2.3 Based on Scope:

- **Global explanation**: explain the overall behavior of the model.
- **Local explanation:** explain each decision or prediction.[11]

5.2.4 Post-hoc Explainability:

- **Explanation by Simplification:** creating a new system that is more straightforward while maintaining the functionality and performance of the trained model to make it explainable and less complex.
- **Explanation by Feature Relevance:** calculate the importance of a feature to the output decision by computing the relevance score for the managed variables using an inner function. (SHAP as an example)
- **Explanation by Visualization** visualizes the model behavior and can be used with other types for improved understanding. (Grad-CAM as an example)
- **Local Explanation:** explain only part of the system by segmenting the solution space and explaining less complex solutions subspaces that are relevant to the whole model.
- **Textual explanation:** generate a textual explanation to help explain the results, mainly used with other types.

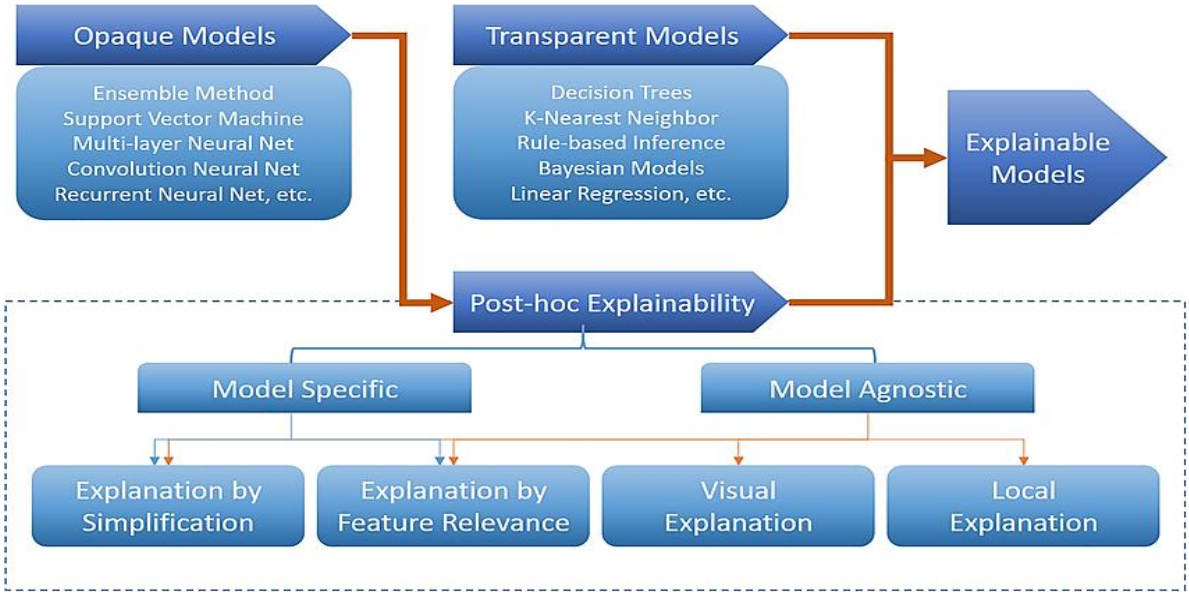


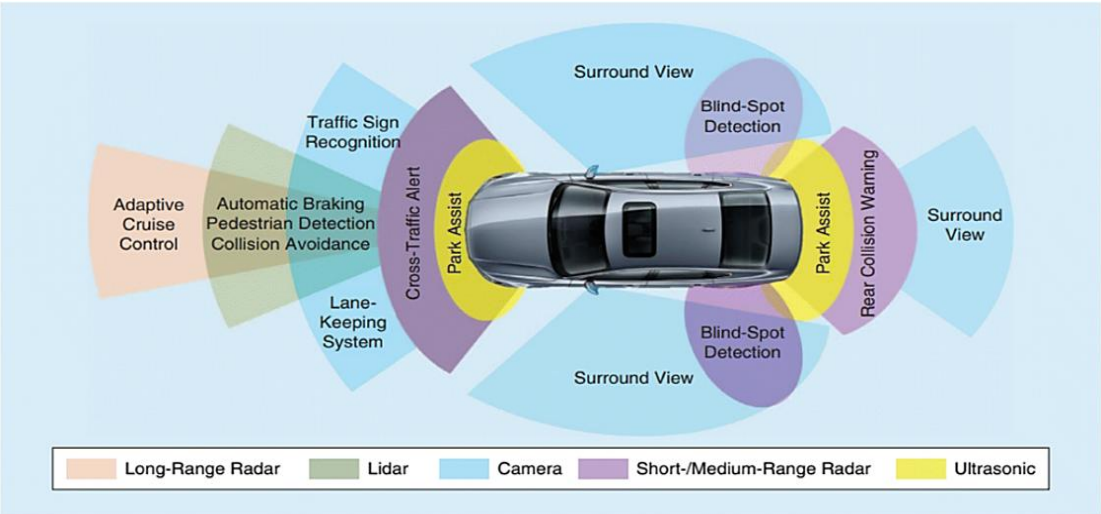
Figure (4): High-level ontology of XAI techniques [6]

Nevertheless, what makes an AI an explainable one? In 2020, the National Institute of Standards and Technology (NIST) published four primary principles [6] for AIs to follow to be considered XAI:

- Explanation: The AI must provide proof, support, and reason for its decision-making.
- Meaningfulness: the explanations provided by the AI must be understandable and meaningful to the stakeholders, and with different stakeholders, each having their own experience and explanations, the explanations need to meet the stakeholders' needs.
- Accuracy: The explanation must be accurate to the AI's decision and processes.
- Knowledge limits: the AI explanation operates by its design and must inform if it is an edge case or if its answer is unreliable.

These reviews provided insight into how AV works, their perception, planning, and control tasks, and the sensors used. [4], [12], [13], [14], and here is what we learned.

Autonomous vehicles use a combination of sensors for their perception, such as Cameras, Light Detector and Ranging (LiDAR), Radio Detection and Ranging (RADAR), and Ultrasonic Sensors (US), each providing unique data about the environment, which we will show in Error! Reference source not found., with cameras being excellent for object detection and signs recognition, but struggle if there is not enough light, LiDAR is used to provide a 3D point cloud data but get obscured in difficult weather conditions like rain or snow, RADAR can determine the velocity of an object but cannot know what the shape of the object, and the US used in detecting obstacles but lack range, by integrating these data in a sensory fusion creating a clear and accurate representation of the environment using each sensors data while eliminating their weaknesses [14]. With 6 representing a real autonomous



vehicle for Waymo company.

Figure (3): Sensors used in autonomous vehicles [13]

These data are used in the perception stage and extracted for information for a variety of tasks, including 2D object detection for recognizing dynamic objects in the environment (signs, traffic lights, pedestrians, and other vehicles), which is the primary method for detection, 3D object detection by adding depth into 2D objects to convert them, object tracking for detecting object trajectory and velocity and knowing where objects are moving, image segmentation for separate objects with different trajectory and object that are not suitable for bounding box such as sidewalks and traffic lines, and road and lane detection that detecting where are the drivable surfaces [12].

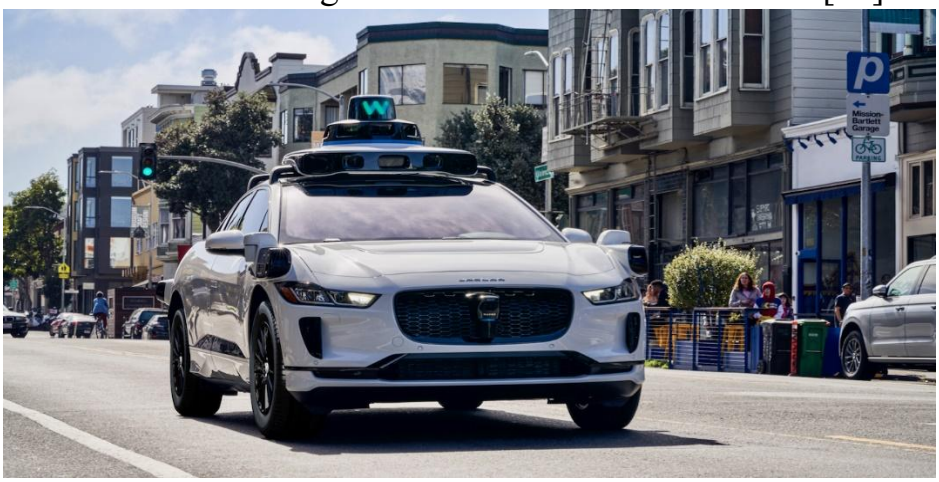


Figure (6): Waymo self-driving car [56]

Current AV differ in their automation level that are based on the technology and AI that they use, with the Society of Automotive Engineers (SAE) [15] diffing six different levels of autonomy defined in their standard J3016, with level ranging from 0 with no automation to 5 full automation, as shown in 7 with level 0 (no automation) the control of the vehicle is solely done by a human driver in all it responsibility including acceleration, braking, and steering with no automation, examples of that are the lane departure warning

or forward collision warning, where the intention is only to warn the human driver without doing the task, level 1 (driver assistance) the vehicle can automate some feature such as steering or acceleration but not both simultaneously, an example of that is the adaptive cruise control that can maintain the speed based on the vehicle in front, level 2 (partial automation) the vehicle can now control both steering and acceleration/deceleration simultaneously under specific conditions but the driver must remain engaged, examples of that are the lane-keeping and adaptive cruise control, level 3 (conditional automation) the vehicle can control all the driving actions with no human intervention, but must still be paying attention in unexpected cases for him to take control over, level 4 (high automation) the vehicle can fully take control over all driving tasks even navigating complex city traffic and in all weather conditions with no human supervision required in a predefined operational design domain, level 5 (full automation) the highest level automation with the vehicle capability of doing all the driving tasks under any conditions anywhere with no requirement for a human driver [15].

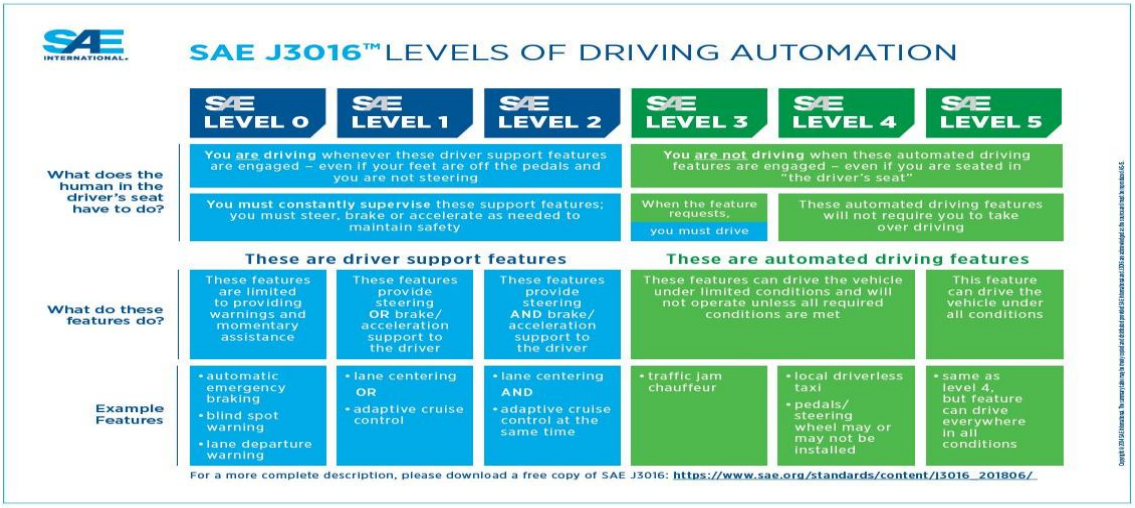


Figure (7): Levels of automation [15]

For XAI in AVs, all these reviews [7], [16], [17] reviewed recent literature on different object detector models and interpreted them, and [14] the book provided an overview of autonomous vehicles and smart cities and XAI, but also provided knowledge that we discussed earlier about XAI and AV.

Object detection is responsible for two tasks: the first is localization, to locate where the object is in the image, and the second is classification, classifying the detected object into a class such as pedestrian, vehicle, or traffic light. There are two types of object detection: two-stage and single-stage, two-stage as the name suggests, they have two stages; in the first stage, they propose various Regions of Interest (RoIs) in the image that have a high likelihood of containing an object, and in the second stage the RoIs that are the most promising are selected with the other discarded, but in single-stage, it uses a single feed-forward neural network to make bounding boxes around each object and classify them in the same stage [18] with 8 illustrates the differences. Two-stage detectors are more accurate, while single-stage detectors offer faster detection; therefore, we must have a delicate balance between speed and accuracy [19].

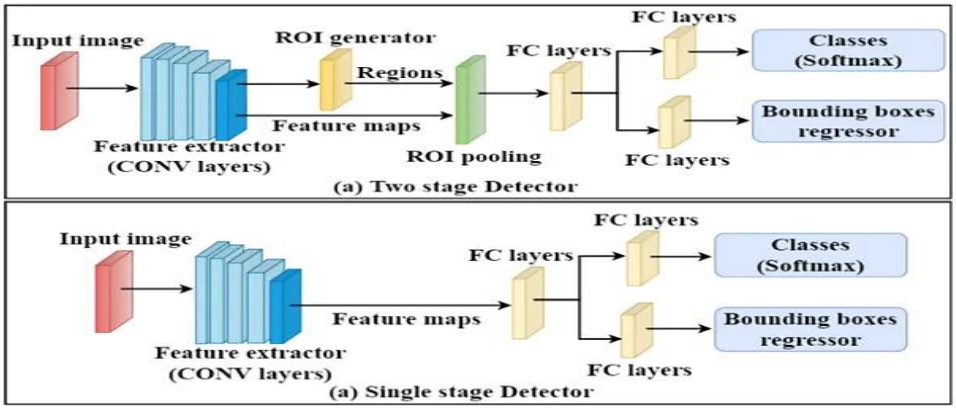


Figure (8): Two-stage vs Single-stage detectors [18]

For object detection [20] showcased the timeline of object detectors and the improvements that were made in the last 20 years, and [18] provided a clear picture of the difference between two-stage and single-stage detectors, while [19] compared the two, overall, through all the papers we gained valuable information about object detection.

5.3 Overview of Related Work:

In various studies the concept of casual inference or explanation in action-scene format using textual explanations (e.g. "stop" to the scene "pedestrian in front"), with Jinkyu K. in his work [21], Introducing an advisable and explainable driving model that incorporate human advice, the model have 4 main component: visual encoder extract visual representation using ConvNet highlighting critical areas that influence the vehicle decision and generating an object centric attention map using DeepLabv3 semantic segmentation that provide a pixel-wise segment that identify object boundaries and Mask R-CNN for more refined attention maps, a vehicle controller take visual and human advice which it convert them into a high level command, it uses two types of Long Short-Term Memory (LSTM) the first Control LSTM that perform the control commands and the second is an advice LSTM that uses the human advice, an observation generator produce textual description/explanation about its visual observation as a pair with the vehicle action and the explanation for the action, the sequence-to-sequence observation-to-action generate a condition-action rule based on the mapping of textual description (e.g. "there is sharp turn ahead") to the high-level command (e.g. "slow down") which gives the model situational cases. The research uses an attention map highlighting the area that influenced the vehicle's decision and textual explanations for its actions based on visual observations. The research used the Berkeley DeepDrive eXplanation (BDD-X) datasets based on a typical BDD dataset containing videos from vehicles but with an added action description and

justification. It was evaluated using Average Placement Error for performance, BLEU, METEOR, CIDEr-D, and SPICE for text, and the model outperforms prior work [22].

With another study by Yiran Xu, with similar concept implemented a multi-task CNN architecture for predicting action and their explanation, this time the paper focuses on action-inducting object, objects that influence the driver decisions, the proposed multi-task CNN predict the action and the explanation using the object detection of Faster R-CNN and global scene context for global feature and local feature providing the overall scene context, combining them into an object-scene relationship (e.g. stop because pedestrians are crossing the street), the paper also created a dataset called Berkeley DeepDrive-object induced actions BDD-OIA based on the BDD100K datasets data with 5 pedestrians or bicycles and more than 5 vehicles, the result show that the model is evaluated against Resnet as a baseline and another research approach with F1 score being the primary evaluation metrics with Resnet performing the worst, other model is better but the proposed model outperformed with it integration of both local and global reasoning, which showed that the explanation can improve on the interpretability and the performance [23].

Chengxi L. proposes a driver-centric risk assessment framework that identifies risk objects as objects that impact the driver's action, such as pedestrians and vehicles, in a cause-and-effect way; the framework contains a two-stage risk object identification. The first stage is an object-level manipulable driving model, the second is causal inference for risk object identification. Object-level manipulable driving model is trained to predict driver action and uses Mask R-CNN and Deep SORT for object detection and tracking them. At the same time, RoIAlign extracts object features and causal inference for risk object detection, which re-predicts the driver action while masking objects one at a time, simulating their removal and seeing if its behavior changes; the object with the most substantial impact on the

behavior change is the risk object. The authors evaluated the model against baseline methods like driver attention prediction and object-level attention selector, and the framework achieved a 7.5% increase in performance. The paper uses the Honda Research Institute Driving Dataset (HDD) [24].

Other studies use semantic segmentation to help with object detection, but different XAI techniques, such as those used by Arsh M. in his work [25]. presenting OD-XAI an explainable AI framework for explainable and transparent deep learning for pixel-wise road detection and segmentation tasks used in autonomous vehicles, integrating XAI techniques to interpret the semantic segmentation prediction, having transparency with knowing that model identifies the areas in an image that contributed the most to the prediction and explaining the internal model working, three model were trained and compared in the semantic segmentation model ResNet-18 and ResNet-50 both are residual network unlike normal neural network where the node output is feed only into the next layer nodes as input and having layers that are too deep learning will slow down creating the vanishing gradient problem, residual network node output feed into the next layer and further layers in the network as input solving that problem, with the difference between the two is that ResNet-18 is smaller in depth with fewer parameter but faster than ResNet-50 that may capture more details, and SegNet inspired by VGG-16 is an encoder-decoder layers style where the 13 encoder layers reduces parameters and the corresponding decoder layers upsample expanding the spatial size for high resolution feature map, the last layer apply softmax activation that classify each pixel, for explainability the study used Grad-CAM and saliency map techniques applying it for each model, GradCAM is a backpropagation method that generate a heatmap that highlights the areas with most influence on the prediction, with the last layer computes, giving a visualization interpretation of what are roads and what are not, with saliency map is also a visualization tool that highlighting the

areas that the model deem distinct features to localize the image, and the brightness of the pixel is how much impact it have on the model prediction, the paper used the KITTI dataset that contain images of roads, the images were also preprocessed with reducing the image size for computational demand and brightness, and hue adjustment. The result that is based on Intersection Over Union (IoU) and accuracy as metrics shows that ResNet-18 shows the highest IoU of both training and test set, accuracy and inference time, with SegNet being the second in both IoU and accuracy but lacking in inference time, Grad-CAM and saliency map showed great interpretability visually, validating the prediction and confirming by highlighting the areas that influenced the decision.

And Vani S. Which proposed a dehazing algorithm called Adaptive Dehazing (AD) that uses Haze Level Discriminator (HLD) and a dehazing algorithm Important Dark Channel Prior (IDCP) a prior work of the author [26], to enhance the performance of semantic segmentation in autonomous vehicles under hazy conditions such as fog or smoke, and incorporation XAI techniques for visualization of feature contributions and explanation and interpretation of the algorithms, the AD scheme work in stages the first HLD assesses the haziness of the images into four categories (heavy, moderate, slight, and clear) with image with unacceptable haze levels are flagged, then the dehazing algorithm using IDCP address four issues with dehazing including estimation of atmospheric light, estimation of transmission map, handling sky/non-sky region, and halos, for semantic segmentation it uses DeepLabv3+ and ResNet-101 as a backbone, and finally the use of Grad-CAM as visualization XAI techniques that explain what features are contributing to the classification as a high gradient in the final score, for evaluating the paper also introduces a synthetic hazy image generation which generate a various levels of haziness scenes to evaluate AD and segmentation on, for the creation of the synthetic haze image the Cambridge-driving labeled video

dataset (CamVid) were used, and both Qualitative and quantitative metrics, for qualitive analysis four test images were processed and visualized by Grad-CAM, and for quantitative Intersection over Union (IoU) and pixel accuracy were used, and a comparison between other dehazing method like base IDCP and GCAN, and the result show AD outperforming both of them in IoU and pixel accuracy and having the highest mean IoU, and the effect of Grad-CAM in enhancing feature detection in AD [27].

The rest of the literature uses various models famous for their real-time object detection (YOLO, R-CNN models, ResNets) and applies different explainability techniques. Mert K. also used semantic segmentation, proposing the model-agnostic framework for plausibilization for object detection in autonomous vehicles achieving interpretability, robustness, flexibility and accessibility using Concept Bottleneck Models (CBMs) and Color-Invariant Convolution (CIConv), this framework can detect false positives and hallucinated object, and providing interpretable verification of the detection, the framework is a multi-step process for verification on the plausibility of detecting objects, first the bounding box (BBox) generate a bounding box on each object and cropped from the original image, second the CIConv that is a physics-based color invariance techniques that transform the image using CIConv filters into a color-invariant representation this will make the model more robust as it decreases the reliance on color making it focus more on the geometric shape rather than the color, third the CBM processes the output from the CIConv in two stages concept extraction using multi-label binary classification and identifying predefined concepts producing a presence score for each object, then multi-class classification is feed with the scores into the object class, with the intermediate concept layer output providing interpretability for the object that contributed to the classification, then plausibility check wither the prediction of the CBM is matches the original prediction if not it well be

flagged as a possible false positive, this will ensure each object detection's plausibility as CBM is only monitoring it making it and the original model independent, this framework was tested on two models the first is a YOLOv5 model trained with the COCO dataset, and the second is a SqueezeDet model trained on KITTI dataset, with each model provided with divers environment scenarios to cover for the testing which focuses on identifying hallucinated object where a detected object does not exist, and false positives that also include hallucinated objects and localization error where the BBox doesn't fully overlap with the true BBox, the result presented that the CIConv-CBM accuracy is similar to vanilla CBM but in cases where the images are corrupted vanilla CBM dropped in accuracy below 40% while CIConv-CBM maintained it accuracy with small drops these result showcases the framework robustness, for the models YOLO had a 4% hallucinated objects with 20% precision in the pedestrian class, and 12% hallucinated objects with 35% in the car class, and a fine-tuned SqueezeDet having 56% false positive recall and 81% accuracy in cars and 95% and 72% accuracy in pedestrian, overall the framework resulted in great insight into the detection highlighting the object contribution with CIConv, decreasing false positives and hallucinated object [28].

Jonas H. in [29]. working on YOLO only, introducing Surrogate Object Detection Explainer (SODEx) a model-agnostic for explaining object detection using Local Interpretable Model-Agnostic Explanations (LIME) framework, aiming for transparency and interpretability in object detection, the model used YOLOv4 a one-stage object detector that uses bounding boxes and being fast and accurate suitable for real-time processing, and the use of LIME for explainability, a model-agnostic local explainer that interpret the complex YOLO model single predictions by using a simpler surrogate model, LIME segment the image into superpixels or regions of the image and explained how each superpixel contribute to the model

detection, and Surrogate Binary Classifier (SBC) that for each object detected from YOLO it focuses on a single detection prediction and compute the Intersection over Union (IoU) between the detected object and the object under explanation and the class score generated by YOLO for how much does it believe in the detection, the SODEX then combine both LIME and SBC taking the class score from SBC and using LIME for local explanation of which superpixel contributed to the detection of each object detected by YOLO creating an explanation, to train the YOLO model the COCO dataset were used, a qualitative and quantitative evaluation were made with qualitative showing that explanation focused on areas that is inside the detected object bounding box, and that legs and arms were influential in detecting peoples, the experiment generated for different types of images and visualizations of it important areas, for quantitative it was evaluated using 2 metrics in-out importance ratio (iratio) that measure how important the pixels inside the bounding box to the ones that are outside wither positively or negatively, and in-out weight difference (wdiff) that calculate the average weight difference of inside minus outside, the result demonstrated the impact that SODEx provide with it accurate explanation for YOLO detection, highlighting the key areas for the detection, the iration and wdiff metrics showed that majority of object detected by YOLO focused more on pixels inside the bounding box rather than the surrounding areas, showcasing that the detection focuses on the object itself. The other literature works on multiple models.

While Majed A. Presenting XAI-SALAPAD an explainable model for detecting and character recognition for Saudi Arabian license plates using a deep learning model, which will assist the Intelligent Transportation System (ITS) with Automatic License Plate Recognition (ALPR) being more accurate and adding interpretability, in monitoring, security, and traffic management, this study combined both You Only Look Once (YOLO) for detecting the license plates and

a convolutional neural network (CNN) for recognizing the alphanumeric that is on the plates, while using Local Interpretable Model-Agnostic Explanations (LIME) for explainability and interpretability, the XAI-SALAPAD is built in four stage data preprocessing, feature extraction, classification, and recognition, for data preparing and processing it the created dataset is comprised of 593 images from internet and phones containing a Saudi Arabian license plates with both Arabic and English alphabet and numbers, in preprocessing step1 is bilinear interpolation for make images the same size, step2 is image binarization for black and white image, step3 is blemish reduction that remove any horizontal and vertical lines, step4 is removing small objects to reduce noise, step5 is cropping the unrelated areas like logo and "KSA" word, for plate detection we use the YOLOv8 for it efficiency, real-time processing, and accuracy with added element such as anchor boxes, intersection over Union (IoU) thresholds, and NMS, for characters recognition we used CNN model using the detected plates from YOLO, the model consist of two fully connected (FC) layer and SoftMax output layer for ability to solve multi-class classification efficiently, and the use of LIME in explainability providing transparency through highlighting features that influence the detection the most in the image, the models were evaluated using the usual metrics of precision, recall, mean precision, and mean average precision (mAP) and the result showcase that YOLOv8 archived mAP@0.5 of 96% in the test set (5% of the dataset) and mAP@0.95 of 97% in the training set (95% of the dataset) outperforming prior YOLO versions in both precision and recall, CNN also achieved good results with 94% accuracy and an F1 score of 93%, LIME illustrated that YOLO focuses on distinct features of the plate like characters edges and that the lower parts of the plate have negative effect on detection due to noise and angle variations.[30]

Maliad M. Introduces the Black-box Object Detection Explanation by Masking (BODEM), a model-agnostic explanation method for object detection tasks,

through three stages: in the hierarchical random masking generation stage, it generates a coarse-grained mask to identify the most salient regions of the object in an image at a high-level, while at low-levels fine-grained masks refine the salient estimation within the regions, in the model inquiry stage the masked images are given to the detection model for observing the changes made on the output when a particular input is missing, with that difference the saliency estimation stage estimate the importance of pixels for the detected object using a saliency map generated by an explanation method, these three steps are then repeated several times in high-level masking hierarchy to low levels producing a final saliency map that visualize a heatmap of the important areas within the image. The BODEM explanation method was evaluated against two explanation techniques, D-RISE and LIME, using three metrics: deletion, where pixels of the original image are deleted based on the saliency value, and the difference between the detection result and the image with deleted pixels is measured, insertion where after the deletion of pixels, pixels insert again based on the saliency value and the difference between the detection result and the image calculate the inserted pixels, convergence is the stability of the generation of saliency map by calculating the difference of three saliency map of the same experiment using Euclidean distance, the three explanation techniques were tested on three models with three datasets a user interface detection using YOLO, an airplane detection using R-CNN, and vehicle detection using SSD and COCO dataset. The result showed more accuracy and less noise compared with D-RISE and LIME and how it is more adaptable for the black-box model as it only needs the coordinates of the detected object rather than the other methods where they need more information like probability and objectness score, but the limitation where the computational complexity with generating multiple masks and repeated inquiries and small size of test sets. [31]

Yanfen Li, addressed the challenges of real time object detection an end-to-end deep learning-based hybrid framework for object detection, assessing pedestrian and vehicles intention, and traffic light recognition was introduced, with the emphasis on explainable AI for better decision making and transparency and safety for autonomous vehicles, the proposed framework can detect 10 different categories of objects in the traffic context such as cars, bikes, motor, etc., predicting pedestrians intentions by cognizing their postures, risk assessment for vehicles and the classifications of vehicles that are deemed dangerous based on their movement, and traffic light recognition and classifying the different lights to decide the vehicle movement, for object detection the You Only Look Once version 4 (YOLOv4) model that been modified by removing 8 redundant shortcut connections with layer pruning, over all the modifications reduced parameters by 74% and increase in accuracy by 2.6% and real-time performance compared to the original YOLOv4 model, Part Affinity Fields (PAFs) were used for recognizing pedestrian intentions by generate a skeleton of key points to determine pose features, which then feed into VGG-19 model to classify the pedestrians intentions, for risk assessment the vehicles behavior were classified based on the their barking, turning, and crossing by multiple CNN model such as MobileNet, ResNet, VGGNet, EfficientNet with EfficientNet showing the best performance with distinguishing between dangerous and safe vehicles, MobileNet were also used for detecting traffic lights for its efficient and suitability classifying 8 different signal types with an accuracy of 95.75%, for explainability the Random Input Sampling for Explanation (RISE) where used, this method generate a saliency map showing a highlighted areas like taillight, wheels, the colored light in traffic lights, that contributed the most to the classification making the decision transparent and safe. The paper also used the BDD100K dataset, which includes ten types of objects in different scenes and weather conditions, and the Tesla dataset with pedestrians and vehicles. The paper conducted the experiments in a real-world setting, with the

result showing high accuracy in all tasks, with a mean average precision (mAP) of 52.7% in object detection, with PR curves showing improvement, and the skeleton-based intention recognition showed 97.5% accuracy. [32]

Another paper by Michihiro K. Introduced the Fast explanation using Shapley value for Object Detection (FSOD) a model-agnostic that generate SHapley Additive exPlanation (SHAP) framework for explanation and interpretability of object detection, this method builds to address the computational complexity challenge of shapely values, the framework present three approaches, the value function for object detection, unlike image classification were you only need the classification prediction score as the value function in object detection you need both classification and localization of the tragic object so a new value function is defined, the utilizing feature maps as input, given that image size in object detection is large learning image features is complex, that why the framework generate feature map as an output from the object detector backbone which will ensure that the explainer can utilize feature maps from multiple scales, the object-specific explanations, that create a query map with an explanation for each object that been detected, and an explainer model for object detection that is based on UNet architecture, for evaluating the framework a primary object detector model was use that being YOLOv5 a one-stage detector using 2 datasets COCO and VOC and other models including YOLOv3 and a two-stage model Faster R-CNN later to ensure that the framework is model-agnostic and can work with other models, and the metrics were Energy-based Pointing Game (EPG) that assesses that the feature attributions highlight the object precisely, Visual Explanation Accuracy (VEA) measures the overlap between the generated saliency map and the ground truth target object using Intersection over Union (IoU), with area under the curve (AUC) as the evaluation metric, deletion and insertion evaluate the impact of important pixels, with deletion removing important pixel and assessing the model

score, and insertion adding pixels and also assessing the model score, as a result the model showcased adaptability with working with all the tested models with good quality. [33].

Other papers use Faster R-CNN, such as [34] by Caio N. But also using YOLO which delves into the critical realm of enhancing the interpretability of deep neural models specifically for object detection within autonomous driving context while focusing on explaining the bounding box regression, the experimental configurations encompass the training on two Faster RCNN models with ResNet-50 and ResNet-101 as backbones and YOLO model for comparison, the datasets including the KITTI benchmark and Pascal VOC 2012, with KITTI having more objects within the images compared to Pascal VOC 2012, with each Faster R-CNN model being trained on both datasets separately creating 4 different models, but YOLO was only trained on KITTI, the models were evaluated meticulously using performance metrics like mean Average Precision (mAP) and smooth L1 loss, for explainability, a comprehensive comparison is drawn between gradient-based methodologies such as Grad-CAM which generate visual explanation highlighting important feature within the image and guided backpropagation that generate a heatmap with the difference between the two is what it compute the gradient with Grad-CAM being the output and backpropagation being the input, and perturbation-based approaches exemplified by D-RISE where it generate noise and see the changes on the model behavior, were made, with these techniques are strategically employed to offer localized and insightful explanations that shed light on the decision-making processes within these complex models, the noteworthy result from this study focused solely on the visual explanations reveals Grad-CAM in susceptible to noise and spatially sensitive but increasing the convolutional layers enhance the noise propagation, and guided backpropagation heatmaps were dispersed they identified the pixels near the object successfully instead of regions

that impacted the detection making explainability not intuitive, but with D-RISE both experiments generated a saliency map that satisfied the localization and reduced the captured noise, a numerical assessment were made with pointing game metrics evaluate the localization of the saliency map, the result showed that gradient-based outperforming D-rise with localization but underperforming in explainability, overall result shows that D-RISE emerges as a frontrunner having the best performance on both Faster R-CNN and YOLO, furnishing clearer and more human-understandable explanations despite the associated computational overhead.

Tomas P. with his paper [35] also used ResNet models, focusing on the significant of the detection capabilities of camera-based object detection systems utilized in autonomous vehicles (AVs), by identifying the critical influential factors that affect the model decision, employing the SHAP (SHapley Additive exPlanations) explainability technique aspiring to substantially bolster the overall safety of autonomous vehicles, the paper approach encompasses a detailed and rigorous four-step pipeline that leverages meta-information derived from each object and it environment for evaluation, a variety of object detection architectures, including but not limited to Single Shot MultiBox Detector (SSD) a one stage detector with fast proccing, RetinaNet also a one stage detector with balance in both accuracy and speed, and Faster-RCNN a two stage detector with high accuracy but low speed, all finetuned on nuScences dataset providing real-world driving data, and these models are assessed for their performance, using Intersection over Union (IoU) and recall, object are categorized based on their meta-information into groups to access their factoring on the recall score, also leveraging a random forest model that been trained to predict the detection solely based on the meta-information and using TreeExplainer a variant of SHAP for tree based models that will provide both local and global explanations through the Shapley values that

calculate each feature contribution for the detection, and illuminating the significant of meta-information factor such as object category and occlusion, the result showed that Retina achieved the highest accuracy and performance with 69% recall and 40% mAP, and Faster RCNN was slightly behind with 67% recall and 36% mAP, and lastly SSD had the lowest recall 47% and 23% mAP but was the fastest, and the profound impact of several pivotal factors and mete-information, such as the size of the bounding box with larger boxes significantly improved recall, the angle of object rotation like how object facing the camera have higher recall and 20% increase in the object detection, the position of the objects with object near the center having more recall, and their distance from the camera with detection dropping after 30 meter, and the occurrence of occlusion, all of which play a crucial role in influencing the overall detection performance of the systems under examination.

This paper by Tomasz N. used Faster-RCNN to proposing a novel method to improve the object detection in Advanced Driver Assistance Systems (ADAS) using an explainable deep learning technique using Faster R-CNN with attention map visualization for interpretable model, the research focuses on enchantment for object detection for electric bus charging stations and addressing the limited dataset and real-world scenarios challenges, for the object detection the Faster R-CNN model is used, a two-stage detection that uses Region Proposal Network (RPN) for generating bounding box candidates, a classification and regressor processes the bounding box candidates based on the feature extraction that is done by the Inception ResNet v2, a heat map generated with harm colors representing areas with high attention and cool colors are less relevant like background, that is used in detecting false positive cases and identify features that are influence the detection, the data were collected from 2 datasets the PUT dataset with captured images from a real charger that was placed at campus and SBC dataset that contain

real-world images, the result showcased that that the model that was only trained on PUT dataset struggled when tested in real-world environment, for attention map some examples revealed correct detection but incorrect attention focus on the people, so a new dataset was added with the KITTI dataset with negative examples with no bus charger to eliminate false positives. [36]

And Ahmed T. proposing a hybrid XAI framework that combine LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations) for explainable real-time object detection in autonomous vehicles, balancing the efficiency of LIME and the accuracy of SHAP, the framework is tested on multiple neural network models such as ResNet, the framework consist of three main component, the perception and decision-making module for multi-model sensor fusion processes data from multiple sensors and generate the bounding box for the objects detected using three different model ResNet-18, ResNet-50, and SegNet, and create the driving decision defined as (S,A,T,R) where S is the state or the (environment context), A is the action (the possible driving action), T is the transition (the probability of state moving), and R is the reward (the desirable outcome) that is based on the Markov Decision Process (MDP), the explanation generation module combine LIME that provide efficient but less accurate local explanation and SHAP with accurate but high computational cost, the explainability integration module ensure efficient integration of the explanations generated by the framework into the decision-making process, the KITTI dataset were using in training the different models, for evaluation Intersection over Union (IoU), accuracy, precision, recall, and inference time were used, with fidelity, consistency, and feature importance used as explainability metrics, the result showed improvement of IoU and accuracy ResNet-18 improved from 0.9459 to 0.95 for IoU and 0.9761 to 0.9780 for accuracy, with slight improvement for precision from 0.965 to 0.97 and recall 0.95 to 0.955, and

reduction of interference time from 0.3421 to 0.28 with other model showing similar results, for explanation the proposed model displayed more fidelity across the models by 85%, and consistency by 70%, and outperformed stand-alone LIME with lower computational cost than SHAP in feature importance. [37]

Lastly this paper by Mehdi M. evaluated three different object detection model for autonomous vehicles Support Vector Machine (SVM), You Only Look Once (YOLO), and Single Shot Multibox Detector (SSD) the models are compared in detail in terms of effectiveness for object detecting, their accuracy, and their real-time capabilities, the evaluation is based on certain tasks like object detection, feature extraction, bounding box generation, and performance all measured on a video dataset, first we have linear SVM is a balanced model between it accuracy and computation, and it trains on Histogram Oriented Gradients (HOG) that simplify image by extracting the objects from the background distinguishing them for object detection, it then apply the sliding window techniques to locate objects making the image split into overlapping areas, with false positive object removed in heatmap and bounding boxes, second is YOLO the data are pre-processed with the images being resized and bounding box are coordinated to standard size, and uses Intersection over Union (IoU) to evaluate accuracy, the YOLO model is based on a Convolutional neural network (FCN) pre-trained on ImageNet dataset with 24 convolutional layers, and simultaneous detection of objects through the use of Non-Max Suppression (NMS) and reducing overlapping boxes, the third is SSD that is based on feed-forward CNN with multi-object detection and predefined boxes similar to YOLO, with training that determined default boxes to ground truth and scale and hard negative mining and data augmentation, the data for evaluating was taken from a video from a real driving car. as the evaluation data is taken from a video so the result evaluation is accessed with Frames Per Second (FPS) and accuracy SVM archived 2FPS which make it unsuitable for real-time processing

due it its sliding windows techniques, YOLO achieved the highest speed with 50-70FPS due to its single-shot detection, and SSD close second with 40-60FPS, for accuracy SVM detected 50% of the object with a false positive of 18%, YOLO with 59% with a false positive of 9.23%, and SSD with the highest 81% with a false positive of 4.93%, in conclusion SVM in very slow and unreliable for real-time object detection, YOLO and SSD provided good efficiency with YOLO being the fastest but with worse performance then SSD. [38]

Table (1 shows all the references with a small summary, their model, the applied techniques, and the datasets used.

Table (1): Summary and comparison of literature

Author	Summary	AI model used	XAI method used	Datasets	Ref
Jinkyu K.	An advisable driving model with four main components: visual encoder, vehicle controller, observation generator, and observation-to-action	ConvNet, semantic segmentation, deepLab v3	visual, textual	BDD-X	[21]
Yiran Xu	action-inducing object detection using Faster R-CNN and global and local features, it showed that the explanation improved the performance	Faster R-CNN	Global/local feature, textual/visual	BDD-OIA	[23]
Chengxi L.	An object-level driving model with a two-stage risk object identification using MASK R-CNN and DeepSORT to detect objects and predict the driving action, and remove objects to see if it changes the prediction. The framework increased performance by 7.5%	MASK R-CNN, DeepSORT	Causal inference	HDD	[24]
Harsh M.	a pixel-wise road detection using three semantic segmentation architectures and applying various XAI techniques to it	Semantic segmentation, ResNet-50, ResNet-18, SegNet	Visual explanation, Grad-CAM, Saliency map	KITTI	[25]

Vani S.	An adaptive dehazing scheme for addressing segmentation in hazy environments using semantic segmentation, DeepLab	Deeplab, Resnet-101, semantic segmentation	Grad-CAM	Cambridge-driving Labeled Video dataset (CamVid)	[27]
Mert K.	a model-agnostic plausibility framework for object detection, solving the interpretability problem of the DNN model using concept bottleneck model (CBMs) and Color-invariant Convolution (CIConv)	SqueezeDet, YOLO	Concept Bottleneck Models (CBMs), Color-invariant Convolution (CIConv)	MS COCO, KITTI	[28]
Jonas H.	proposed SODEx, a model-agnostic surrogate model using LIME	YOLOv4	LIME	COCO dataset	[29]
Majed A.	XAI-SALAPAD is an explainable model for detecting and recognizing license plates in Saudi Arabia using YOLO and CNN, respectively, and LIME for explainability	YOLO, CNN	LIME	New datasets with license plates	[30]
Milad M.	A Black-box Object Detection Explanation by Masking (BODEM) explains the method in three stages: hierarchical random masking, model inquiry, and saliency estimation. The method was compared to D-RISE and LIME with three models, YOLO, R-CNN, and SSD, in multiple contexts and datasets.	YOLO, R-CNN, SSD	BODEM, saliency map	COCO	[31]
Yanfen Li	An optimized YOLOv4 model with 74% reduced parameters and 2.6% increased accuracy, and various CNN models for object detection, traffic light recognition, risk assessment, and intention recognition with PAF	YOLOv4, CNN models (MobileNet, ResNet, EfficientNet)	RISE, saliency map	BDD100K and other subsets	[32]
Michihiro K.	a model-agnostic framework for interpretable object detection with the use of SHAP	YOLOv5, YOLOv3, Faster R-CNN	SHAP	COCO, VOC datasets	[33]
Caio N.	Compared Gradient-based and perturbation-based XAI techniques on an object detection model	Faster R-CNN, YOLO	Grad-CAM, guided backpropagation, D-RISE	KITTI/Pascal VOC	[34]

Tomas P.	analyzed and compared three AI models for object detection, and implemented explainability techniques, SHAP, and random forest	SDD, ResNet, Faster R-CNN, Random Forest	TreeExplainer, SHAP	NuScenes	[35]
Tomasz N.	A tool for a better understanding of deep neural networks, in particular, Faster R-CNN, using an attention heat map	Faster R-CNN	attention heat map	PUT (made dataset), SBC, KITTI	[36]
Ahmed T.	proposed a hybrid XAI model-agnostic framework that combined LIME and SHAP	ResNet-18, ResNet-50, SegNet	LIME-SHAP integrated	KITTI	[37]
Mehdi M.	compared different object detection models in autonomous vehicles in accuracy and real-time processing, and performance	YOLO, SDD, SVM	N/A	Video dataset for a real-driving car	[38]

5.4 Theoretical Framework:

The concept of post-hoc XAI or explaining a model that is not inherently interpretable by design, is that using a desired trained model and an explainability module that will explain the model decision, some will only locally explain the model decisions while others will globally explain the whole model, some will work on all models and some need specific models, in Figure 9 we show a general framework on post-hoc XAI.

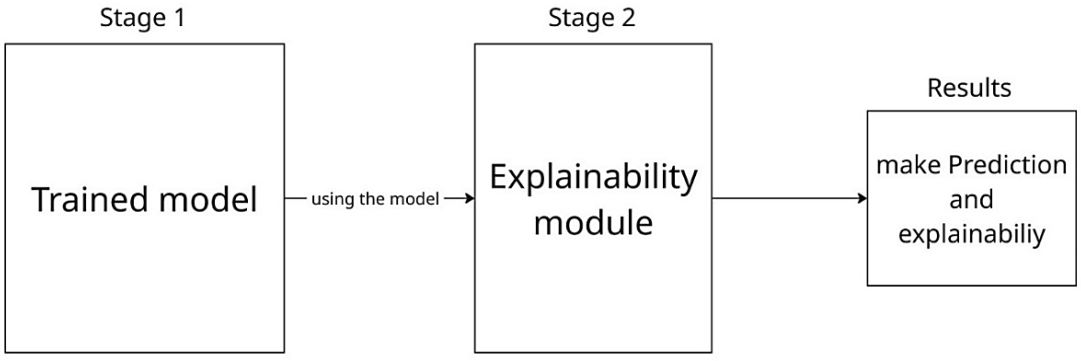


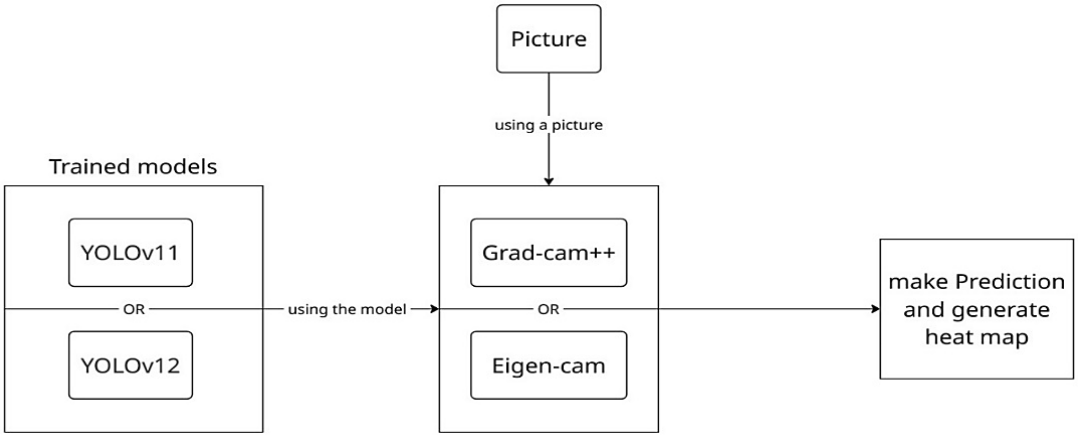
Figure (9): XAI general framework

6. Research Methodology

In this study we will be using experimental methodology to explain the object detection of the YOLO (you only look once) [39] models using Eigen-CAM [40] and Grad-CAM++ [41], we will be designing, developing, and training the YOLO model and implementing the eigen-cam and grad-cam++ explainability techniques to showcase where does the model is focusing. In this chapter, we will discuss the proposed model, how it was built, its architecture, as well as the dataset used.

6.1 System Design/Architecture:

Our proposed system aims to explain the object detection of the YOLO model in autonomous vehicles by using explainable AI techniques to enhance out interpretability of the model, the system contains two components, the object detection models which are YOLOv11 [42] and YOLOv12 [43] the latest version, and the explainability module which uses Eigen-cam, giving us both an accurate detection and we can visually interoperate the model decision making. Error! Reference source not found. shows the processes of the system. Eigen-cam and



Grad-cam++ will use either trained model to predict a picture and generate the heatmap associated with it.

Figure (10): Our proposed system

6.2 Algorithms/Models:

In this report, the experiment will consist of two phases. In the first phase, we will build two YOLO models, YOLO11 and YOLO12, the latest versions of the YOLO [39] model. In the second phase, we will implement the explainability technique, Eigen-CAM, and Grad-CAM++ to explain the detection of the YOLO models.

6.2.1 YOLOv11:

We will start with the YOLO11 model, one of the newer versions of YOLO developed by Ultralytics, which offers multiple applications, from object detection, classification, segmentation, and pose estimation.

As new versions of YOLO, an improvement to the family is expected; the key improvements are the introduction of Cross-stage Partial with Spatial Attention (C2PSA) and replacing the C2f blocks with C3K2 blocks on its architecture, which we will dive into in detail next [44].

YOLO models' architecture consists of three important components. First, the backbones that are used to extract the features using conventional neural networks transform the images into multi-scale feature maps. Second, the neck is the middle stage for processing, aggregating, and enchanting feature representations at different scales. Lastly, the head that predicts and generates the output consists of the classification and the localization of the object using the refined feature maps.

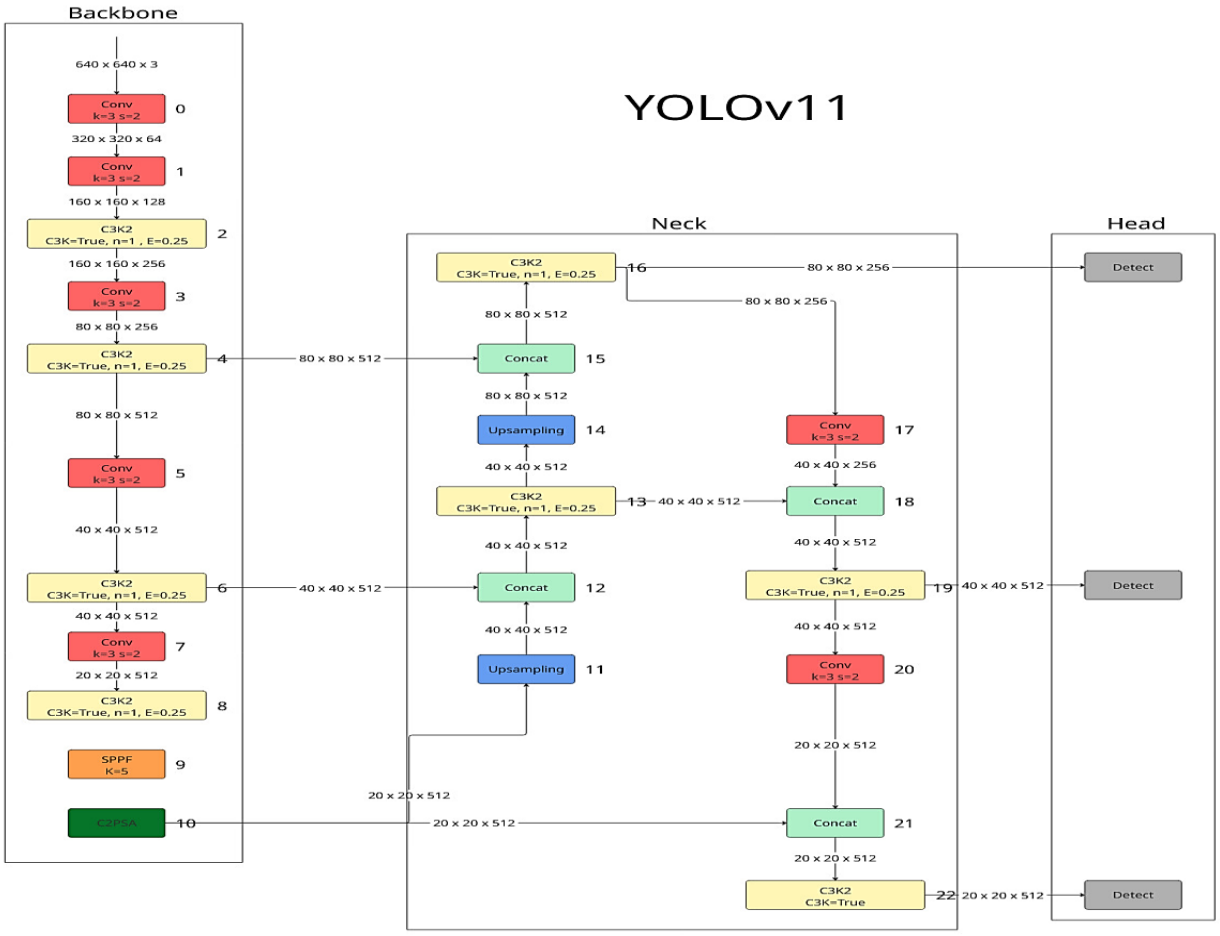
Backbone:

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

One of the three important components of YOLO architecture is where features will be extracted from the input images at different scales using conventional layers and special blocks to generate the feature maps.

Final Convolutional Layers and Detection:

Each detection processes a final Conv2D layer, that reduces the number of features required to output for bounding box and class prediction, the final detection layer combines this prediction to produce a bounding box to localize where the object, objectness score of how confidence of the object presence,



and class score to know that class the object belongs to. This process and all previous ones can be seen in Figure 11, showcasing the YOLOv11 architecture.

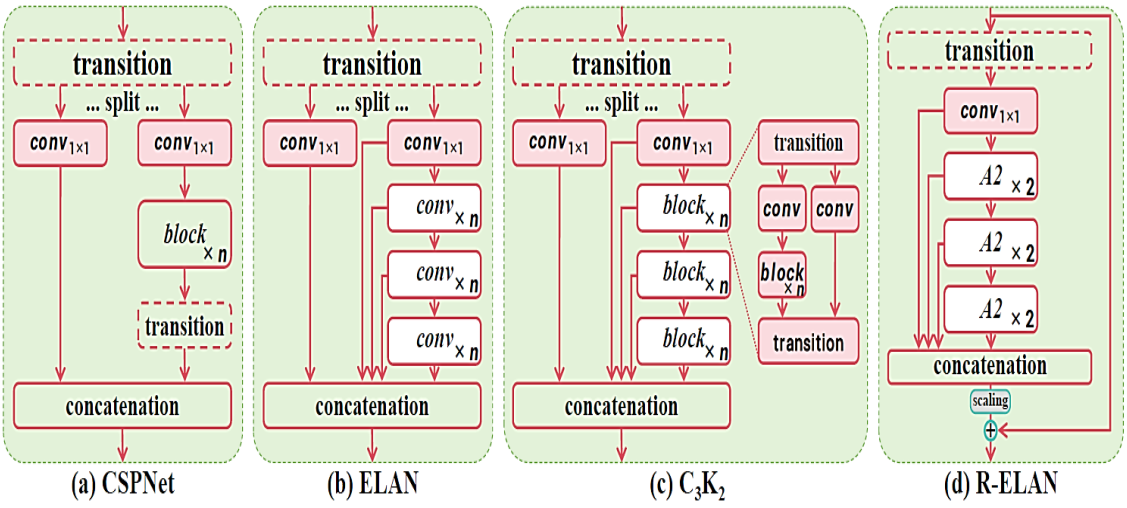
Figure (11): YOLOv11 architecture

6.2.2 YOLOv12:

The newest version of YOLO was released in February 2025, similar to YOLOv11. This version also improved on its predecessor, the key improvements are R-ELAN at the backbone, area attention mechanics at the neck, and refined prediction pathway at the head, which we will go into detail next [46].

Backbone:

Beyond improvement to the conventional blocks, it also implements a 7 x 7 separable convolutions technique to reduce computational resources, replacing the large-kernel convolution operations, ensuring spatial awareness while having fewer parameters, moreover, various sizes of objects such as small or occluded one are represented in the network using the multi-scale feature

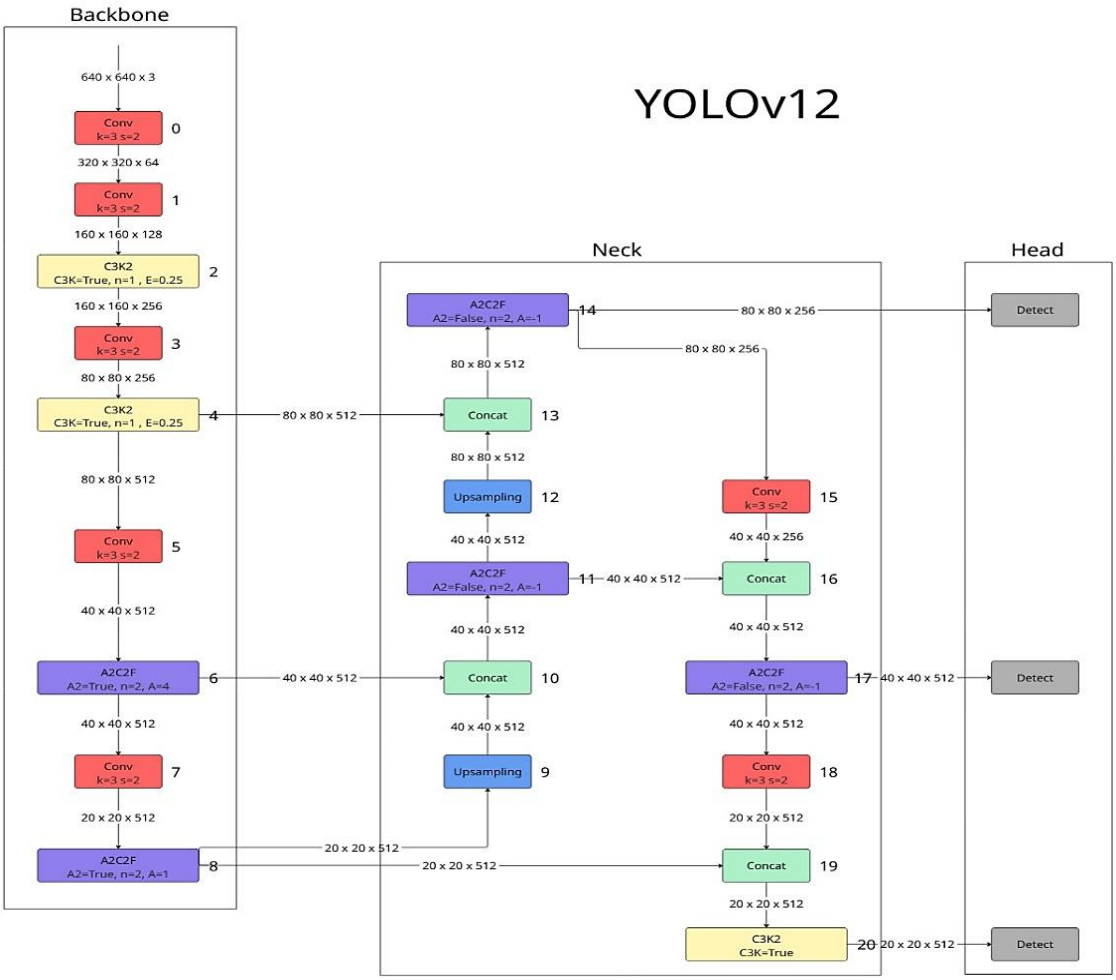


pyramid, with Error! Reference source not found.12 showcasing the differences between YOLOv11 C3K2 and the new YOLOv12 R-ELAN.

Figure (12): The architecture comparison with popular modules. Including YOLOv11 C3K2 and the new YOLOv12 R-ELAN

Neck:

Similar to YOLOv11, the neck aggregates and refines the multi-scale feature for detection, but a key improvement is an area attention mechanism



empowered by FlashAttention, which improves the model's focus on important regions. Error! Reference source not found. showcases a comparison of attention mechanisms and the new area of attention of YOLOv12.

Figure (13): YOLOv12 architecture

Improved loss function that balances the classification and localization of the object, with Error! Reference source not found. showcasing the architecture of YOLOv12.

6.2.3 Eigen-Cam:

A gradient-free visualization technique called Eigen-CAM (Eigen Class Activation Mapping) draws attention to the areas of a picture that have the greatest influence on a model's prediction. It operates by applying Principal Component Analysis (PCA) to the activation maps of the last convolutional layer and does not require backpropagation, in contrast to Grad-CAM. A heatmap highlighting important decision areas is made using the first principal component, which stands for the prevailing pattern. This method helps interpret CNNs since it is more reliable, model-agnostic, and frequently yields simpler explanations.

6.2.4 Grad-Cam++:

Is an improved version of Grad-cam, and unlike Egien-cam, grad-cam/gradcam++ uses gradients of the target class in the final layer to produce a heatmap that showcases the important regions of the image, firstly, in the forward pass it gets the feature maps from the last convolutional layer just like Eigen-cam, but in the backward pass it then computes the gradient of the class score with the feature map and average the gradient and combines them to produce a heatmap. Grad-cam++ improved on the grad-cam limitation when dealing with multiple objects of the same class.

In this study, we used YOLOv11 and YOLOv12 as the newest versions of YOLO and have not been heavily researched, especially in terms of explainability, but why YOLO? There are 2 types of object detectors: one-stage and two-stage detectors. YOLO is a one-stage detector that is much faster and suitable for real-time detection like autonomous vehicles, and the choice of Eigen-CAM and Grad-CAM was for their specialty of convolutional neural networks, and the generation of visual explainability, perfect for object detection, where they generate a heat map showing where it focuses.

6.3 Data Collection:

In this research we will be using two datasets the first is the KITTI autonomous driving benchmark dataset available on their website for free [47], a well-known dataset used for autonomous vehicles, KITTI contains many hours of traffic scenarios using many sensors such as stereo and 3D lasers, in this study we used the 2D object dataset which contains 7481 images with objects such as Car, Cyclist, Pedestrian, Person setting, Misc, Tram, Truck, and Van, the second dataset is Tsinghua-Daimler Cyclist Benchmark also available on their website [48], the data set contain 12014 images of only cyclist in total the dataset contain 19495 images, CityPerson dataset [49] were considered to increase the number of pedestrians but it worsens the results of the model.

We resized all images to 640 x 640 to stander size as the two datasets had different images sizes and to follow the input size for the YOLO model,

Table (2): How many instances per class for each dataset

dataset	Car	Cyclist	Pedestrian	Tram	Truck	Van
KITTI	28742	1569	4487	511	1094	2914
Tsinghua-Daimler Cyclist	-	22173	-	-	-	-

shows how many instances per class for each dataset, instance, where used instead of images per class, is because each image contains multiple objects from the same class and different classes.

Table (2): How many instances per class for each dataset

dataset	Car	Cyclist	Pedestrian	Tram	Truck	Van
KITTI	28742	1569	4487	511	1094	2914
Tsinghua-Daimler Cyclist	-	22173	-	-	-	-

7. Implementation and Experimental Setting

7.1 Implementation Details:

In this section, we talk about the technical setup and tools utilized to create the YOLOv12, YOLOv11 architecture, and Eigen-CAM and grad-CAM-based explainability, including a variety of libraries for image preprocessing, visualization, explainability and support, and cloud-based platforms for training and testing the model using the Python programming language for the implementation of the project.

7.1.1 Programming Language and Framework:

Python: the primary language used for machine learning, deep learning, and data analysis for its extensive collection of libraries that support various parts of the project, and for its ease of use and readability.

PyTorch: one of two famous tools used as a deep learning framework for training, testing, and fine-tuning the YOLO models, which provides tensor and GPU

acceleration support.

Ultralytics: the official open-source library for the implementation of the YOLO family, which includes YOLOv12 and YOLOv11. It offers evaluation scripts, pre-trained models, and simplified training procedures.

7.1.2 Cloud-Based Development Environments:

Google Colab/Kaggle Colab: or the "Collaboratory" method used for providing a suitable environment for researchers, machine learning enthusiasts, and learners of the field, google/Kaggle will provide GPU and RAM for free with better options with paid, this environment can be used to write and modify python codes, training the model for long amount of time to train on large datasets.

7.1.3 Explainability Module:

Pytorch-grad-CAM: the main libraries for CAM, including grad-cam, grad-cam++, and eigen-cam explanation techniques that generate heatmaps that show where the model focuses [50].

OpenCV: For image processing jobs, including resizing, converting formats, and overlaying heatmaps on original photos.

Matplotlib: Used to display CAM heatmaps and detection outputs. Additionally, it facilitated the comparison of raw images, detections, and explanations in a consistent format.

7.1.4 Datasets and Image Preprocessing:

Roboflow: a web-based platform for building, training, and deploying computer vision models. With access to various datasets, we used Roboflow for preprocessing our data and importing it into the development environment.

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

For the implementation we will share the steps used with relevant code, first, we will install the needed libraries shown in figure 14, and then we will import the dataset from Roboflow as shown in figure 15, with figure 16 showing the code for training the model, and in we validate our results using the test set.

```
!pip install ultralytics
!pip install roboflow
!pip install grad-cam
from ultralytics import YOLO
```

Show hidden output

Figure 14: Code containing the installation of important libraries

```
from roboflow import Roboflow
rf = Roboflow(api_key="pegL48guzlGkiTlbCaRc")
project = rf.workspace("osos-workplace").project("object-detection-in-av-rs8lg")
version = project.version(3)
dataset = version.download("yolov12")
```

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in object-detection-in-AV-3 to yolov12:: 100%| | 1025479/1025479 [00:12<00:00, 82771.78it/s]

Extracting Dataset Version Zip to object-detection-in-AV-3 in yolov12:: 100%| | 39002/39002 [00:05<00:00, 7549.43it/s]
```

Figure 15: Code containing import of dataset

International Journal of Computers and Informatics, London Vol (4), No (10), 2025
https://doi.org/10.59992/IJCI.2025.v4n10p1 E-ISSN 2976-9361

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

```
model = YOLO("/content/runs/detect/train/weights/best.pt")  # load a custom model

metrics = model.val(split='test')  # no arguments needed, dataset and settings remembered
metrics.box.map  # map50-95
metrics.box.map50  # map50
metrics.box.map75  # map75
metrics.box.maps  # a list contains map50-95 of each category
```

```
[ ] !yolo detect train data=/content/object-detection-in-AV-3/data.yaml model=yolo12m.pt epochs=120 imgsz=640 batch=42 amp=True patience=50 conf=0.4 optimizer=SGD
Show hidden output
```

Figure 16: Training the model code

7.2 Hardware and Software Environments:

For this research, we utilized cloud-based environments using Google Colab and Kaggle Colab. We used Kaggle in the early to middle stages of development for testing the effectiveness of the model and the datasets, and it was also used for testing and explaining the model with Eigen-cam and grad-cam++. Kaggle had a P100 GPU with 16GB of RAM. While Google was used in training the model, as it had a far better GPU for faster training, the GPU used was the A100 with 40 GB of RAM.

7.3 Experimental Setup:

The final data set contained 19495 images, with 7481 images from KITTI and 12014 from the cyclist dataset, and they were split evenly into a 70% train set with 13647 images, a 20% evaluation set with 3899 images, and a 10% test set with 1949 images.

The parameters used were Stochastic Gradient Descent (SGD) as optimizer, learning rate lr=0.01, momentum=0.9, confidence conf=0.4, amp=True (increase training speed while keeping performance), for both YOLOv11 and YOLOv12,

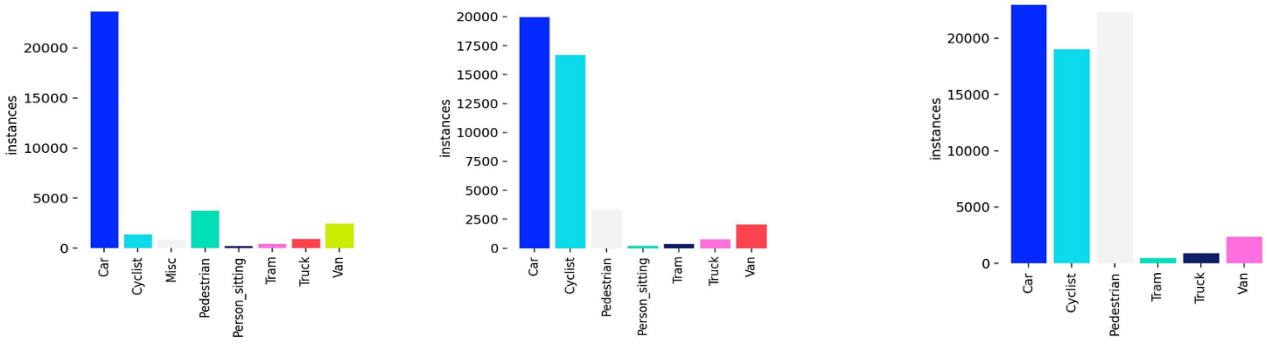
with the difference being YOLOv11 training for 160 epochs with 41 batch size, while YOLOv12 training for 120 epochs with 42 batch size.

Our experiment went through four tested datasets on the YOLOv11 while training the YOLOv12 model on the best dataset. At first, we tested the KITTI dataset only at 32 epochs with unsatisfactory results. Looking at the instances per label in figure 17, A) there was a big imbalance of data; the results of this dataset will be discussed in the results section.

Second, we added the cyclist dataset to solve the imbalance of the cyclist label, which showed good results while training for 40 epochs, with instances per label also seen in Error! Reference source not found. B), but pedestrians still show low results.

Third, we added the CityPerson dataset to fix the imbalance of the pedestrian class, but it had the opposite effect, resulting in lower performance. As we mentioned before, results will be discussed in detail in the results section; the new instance per class can be seen in Error! Reference source not found. C).

Lastly, we removed the CityPerson dataset, and only used KITTI and cyclist datasets to train the YOLOv11 and YOLOv12 models, the imbalance of the dataset is still present and while it could be fixed by adding more data it is hard to find a dataset that only contains the underrepresented classes without also



increasing the car class, and there are also other solutions such as undersampling and oversampling we decided to leave as it is and test the explainability of this model while testing a balanced dataset could be done as a feature work.

Figure (17): instances per label A) KITTI only, B) KITTI and cyclist datasets A dataset, C) KITTI, cyclist, and City Person B C

7.4 Evaluation Metrics:

A variety of common metrics were employed to evaluate the YOLOv11 and YOLOv12 object detection models' performance, concentrating on the correct detection and localization. These metrics will give a thorough picture of the model's performance in identifying objects in practical situations.

7.4.1 Precision:

The percentage of accurately predicted objects among all predicted objects is. It is described as:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

Having high precision means that the model is making few false detections, which is great for object detection as we don't want misleading predictions, but it is not the most important in this experiment, as we have an imbalance of data.

7.4.2 Recall:

The percentage assesses its capacity to accurately detect every true object in an image. It is described as:

$$Recall = \frac{True\ positive}{True\ positive + False\ negative}$$

A high recall indicates that the model misses fewer real things and will be important in our experiment, as some classes have low numbers, so we need the model to detect the majority of them.

7.4.3 F1 Score:

A harmonic means of precision and recall that offers a balanced assessment of performance is helpful when both false positives and false negatives are significant. It is described as:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

When assessing object detection in real-world scenarios, when you have imbalanced data, or when both over-detection and under-detection are expensive, this statistic is especially helpful.

7.4.4 Mean Average Precision (mAP):

The main statistic used to assess object identification algorithms is Mean Average Precision (mAP). It evaluates the model's accuracy in object classification and localization over a range of object classes and confidence criteria.

The average precision is calculated at a specific IoU (Intersection over Union) threshold of 0.50 using the formula mAP@0.50 (mAP@50). This loose criterion determines if the anticipated boxes and the ground truth typically overlap [52].

mAP@0.50:0.95: This more thorough version offers a strict assessment of the model's detection quality by averaging precision across several IoU thresholds (from 0.50 to 0.95 in 0.05 increments).

Following training on the KITTI dataset, these metrics were calculated using the integrated evaluation tools offered by the Ultralytics YOLOv11 and YOLOv12 frameworks.

For explainability, as the Eigen-cam and Grad-cam generate a heatmap with areas of focus from the model, it will be evaluated visually.

8. Results and Discussion

8.1 Analysis of Results:

We will now share the results from our four experiments that we talked about in the experiment details, the first experiment was done on the KITTI dataset alone, the model achieved 0.89 precision, 0.85 recall, 0.92 mAP@50, and 0.7 mAP@50-95 overall showing average results but the confusion matrix in Figure

18 and F1 curve in figure 19 and recall curve in Figure 20 shows underwhelming performance from cyclist and pedestrian classes.

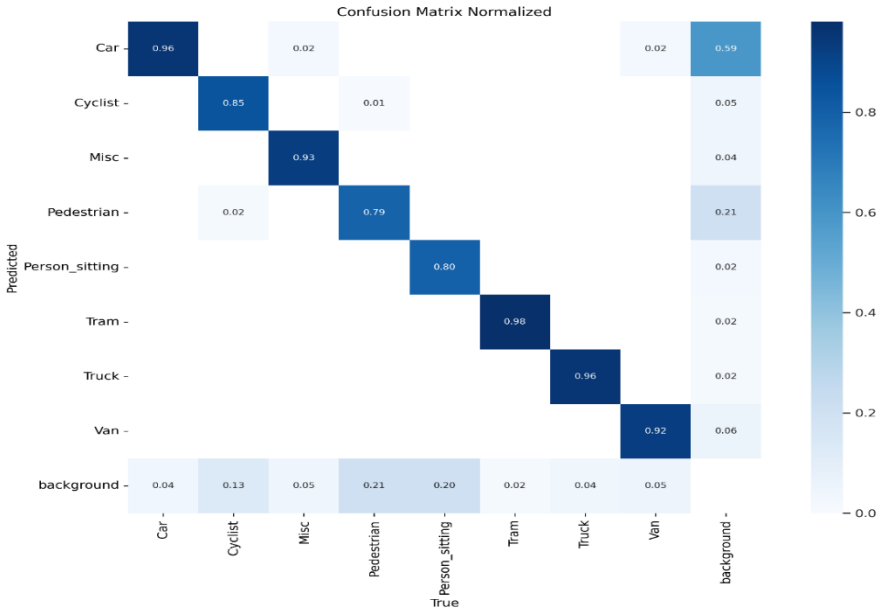


Figure (18): Confusion matrix for the first experiment

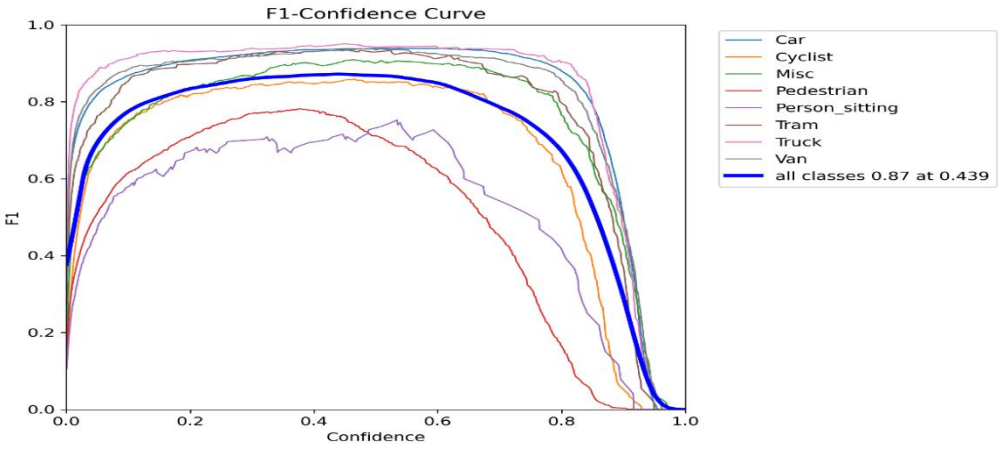


Figure (19): F1 curve for the first experiment

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

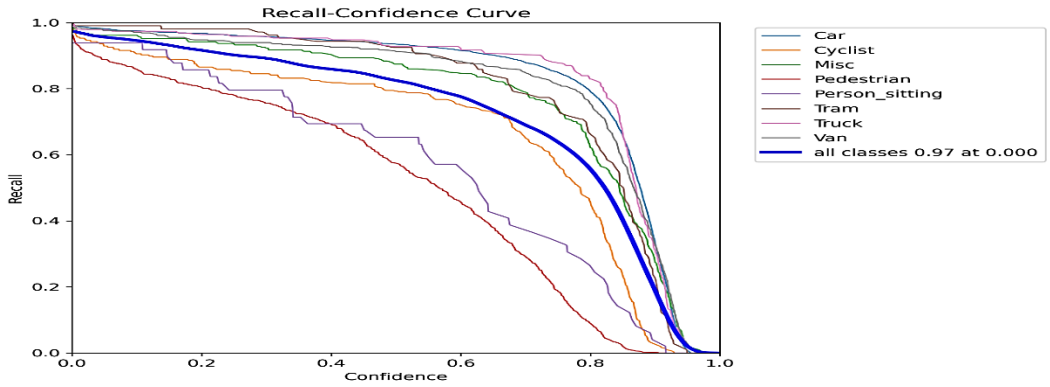


Figure (20): Recall curve for the first experiment

The second experiment showed improved performance for the cyclist class for the addition of the cyclist dataset, with 0.9 precision, 0.84 recall, 0.9 mAP@50, and 0.7 mAP@50-95 overall similar results from the first experiment but the confusion matrix in Figure 21, F1 curve in Figure 22, and recall curve in Figure 23, shows better performance from cyclist class.

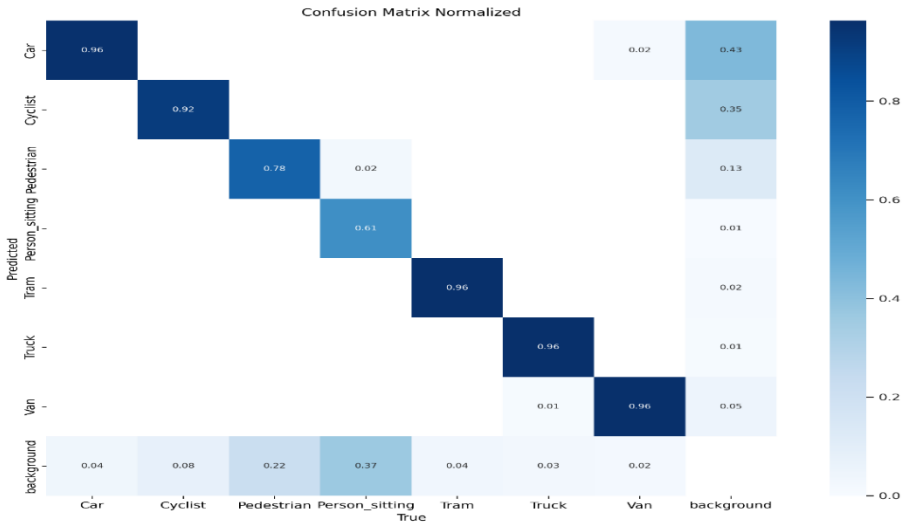


Figure (21): Confusion matrix for the second experiment

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

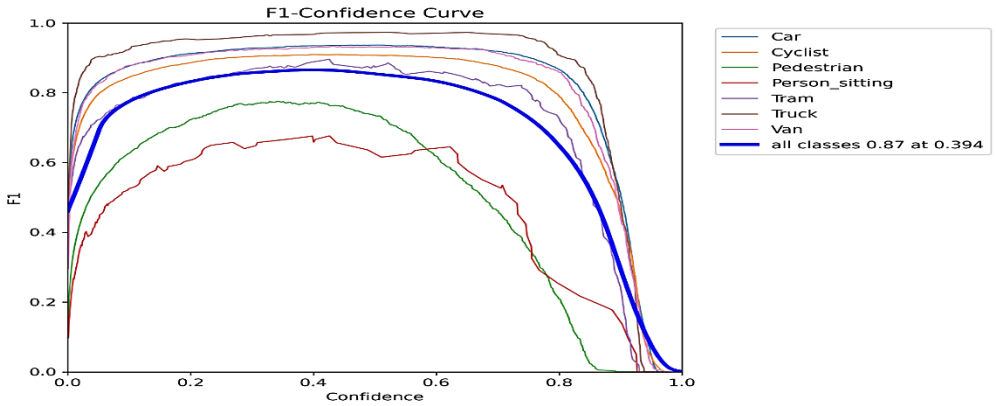


Figure (22): F1 curve for the second experiment

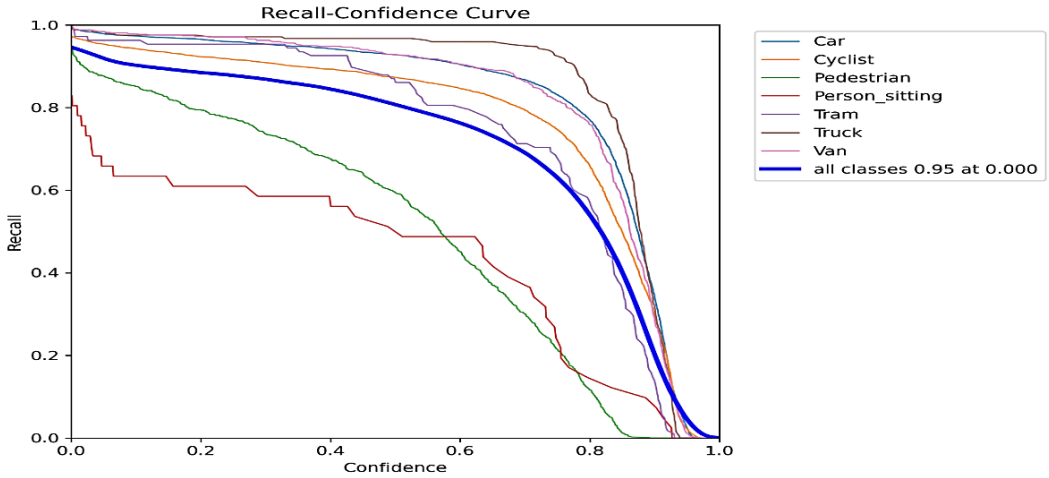
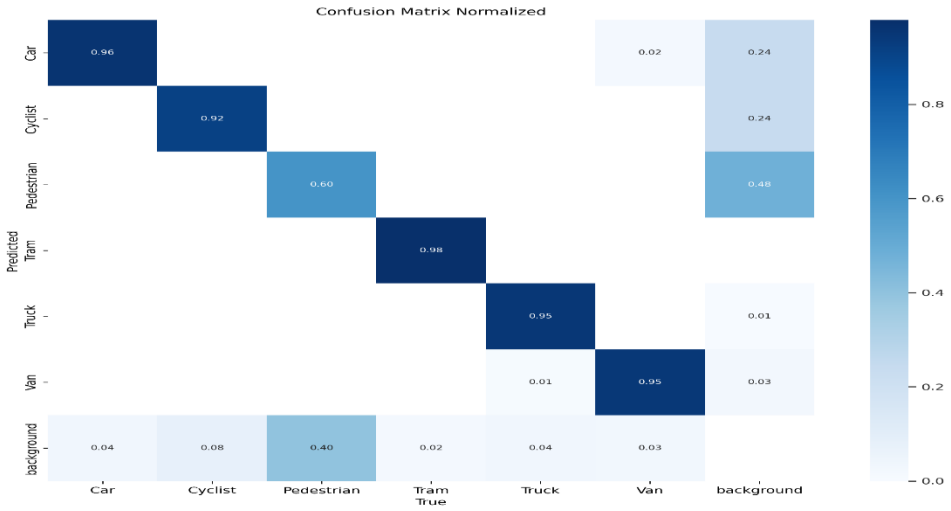


Figure (23): Recall curve for the second experiment

In the third experiment we added the CityPerson dataset to fill the class imbalance of pedestrian class archiving better overall results with 0.92 precision, 0.86 recall, 0.92 mAP@50, and 0.73 mAP@50-95 but the confusion matrix in Error! Reference source not found.24, F1 curve in Figure 25, and recall curve in Error! Reference source not found. 26, shows that pedestrian performance degraded from previous experiments, Figure 27 shows an example of the

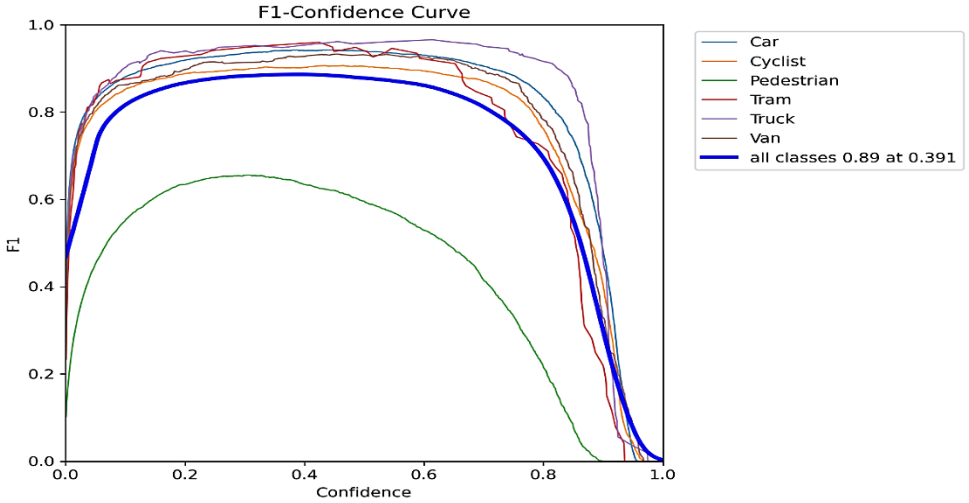
validation batch from the dataset A) contained the actual labels which include labels for small objects and overlapping label that confuse the model prediction



in B).

Figure (24): Confusion matrix for the third experiment

Figure (25): F1 curve for the third experiment



International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

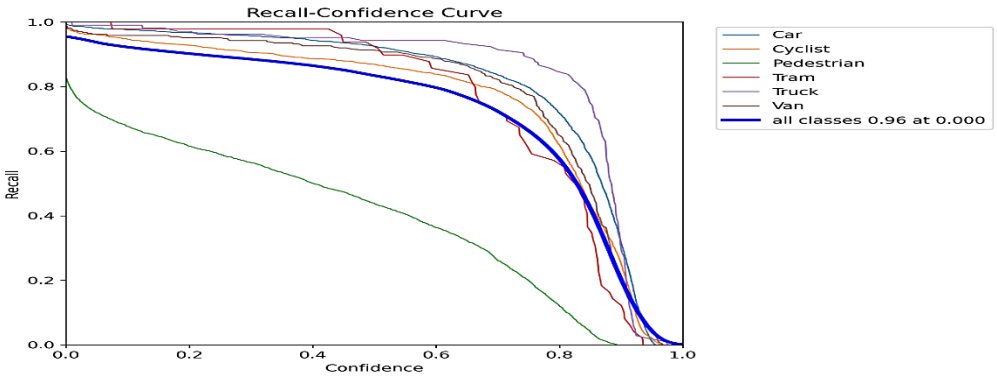


Figure (26): Recall curve for the third experiment

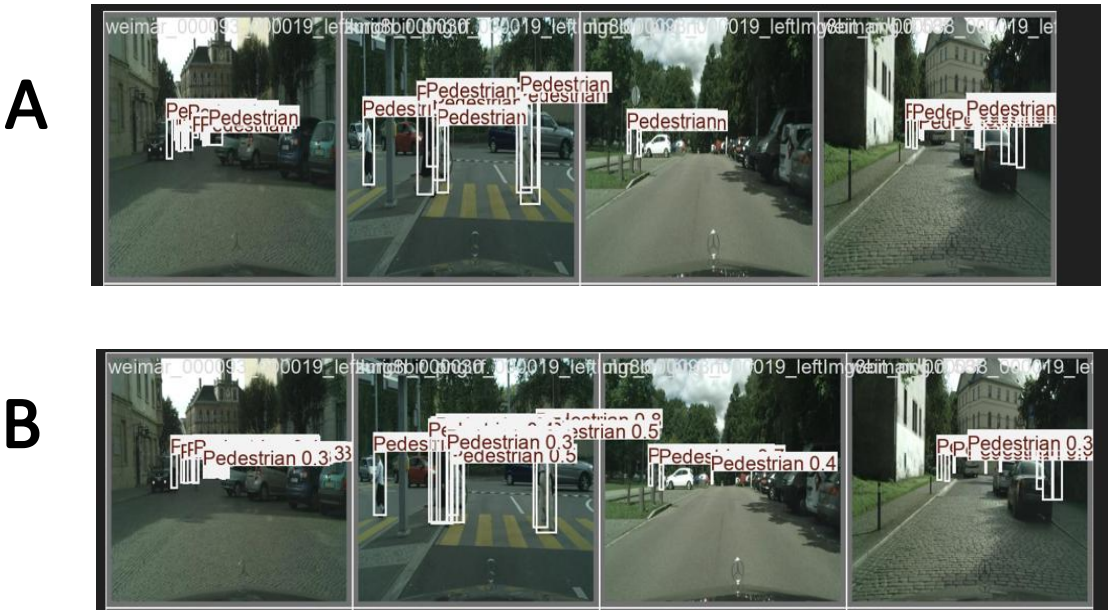


Figure (27): Example from CityPeron validation patch A) contains the actual labels in the image, B) contains the prediction

Lastly, the fourth experiment is the last experiment where we removed the CityPerson dataset, and we trained our YOLOv11 and YOLOv12 on the KITTI and cyclist datasets.

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

8.1.1 YOLOv11:

In our experiment for YOLOv11, we trained it for 160 epochs the model achieved 0.93 precision, 0.93 recall, 0.95 mAP@50, and 0.83 mAP@50-95, the confusion matrix in Error! Reference source not found. 28, F1 curve in Error! Reference source not found. 29, recall curve in Error! Reference source not found. 30, and even precision-recall curve in Error! Reference source not found.31 shows better performance from prior experiments there still underrepresenting on the pedestrian class which have the worst performance, other such as precision cure in Figure 32 show a normal gradually increase in precision for all class, and results in Figure 33 show decrease of box_loss which mean the error when localizing the object, cls_loss which mean the error when classifying, and dfl_loss which mean the error when differentiating between similar objects, as the model train they decrease show normal training behavior. We can see the metrics for each class in

Figure (32): Precision curve for YOLOv11

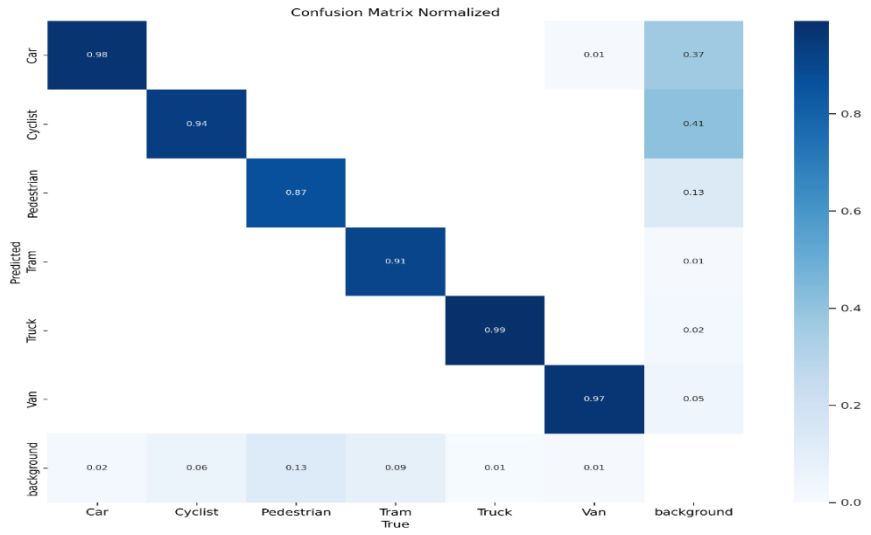


Figure (33): Training result for YOLOv11

Table (3. The seen pedestrian class shows the underrepresentation, having a low recall for false negatives. We can also see that mAP@50-95 is low, meaning that the model can't precisely localize them.

Figure (28): Confusion matrix for YOLOv11

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

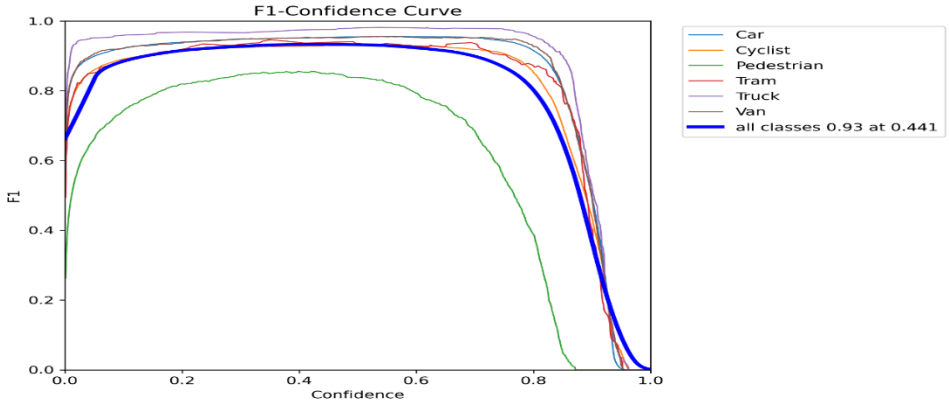


Figure (29): F1 curve for YOLOv11

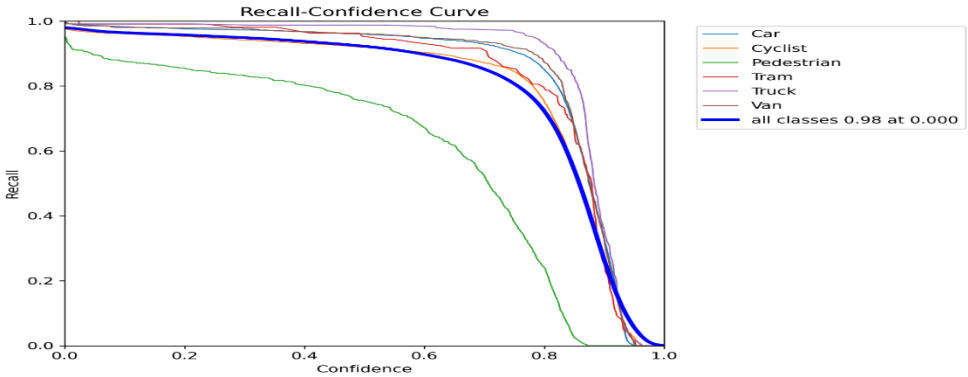


Figure (30): Recall curve for YOLOv11

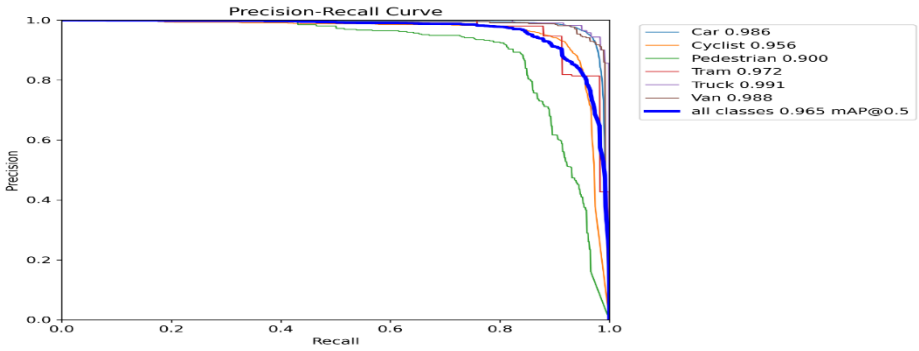


Figure (31): Precision-Recall for YOLOv11

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

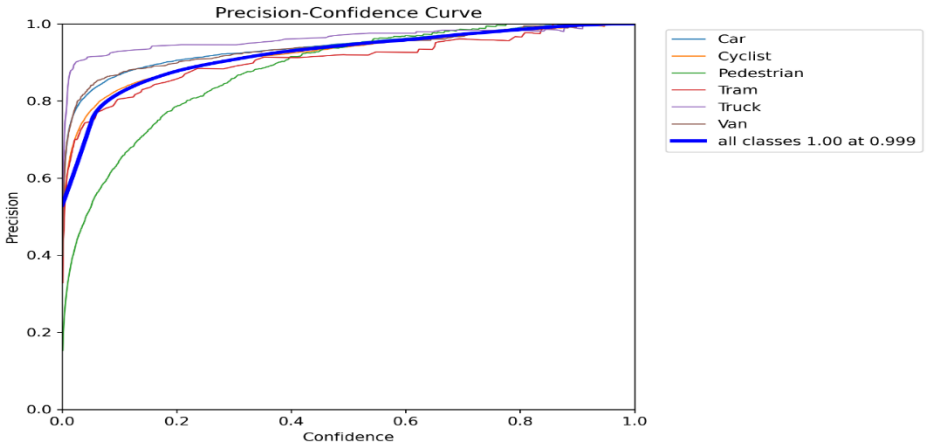


Figure (32): Precision curve for YOLOv11

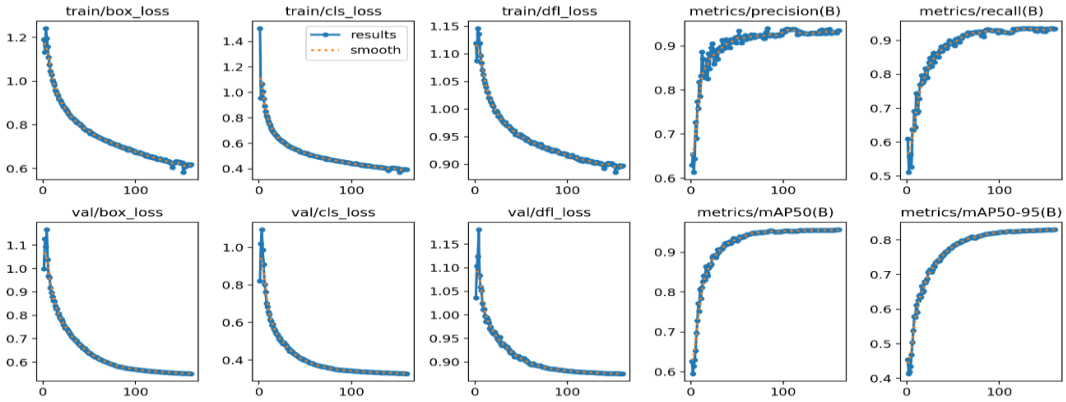


Figure (33): Training result for YOLOv11

Table (3): YOLOv11 metrics for each class

	Precision	Recall	F1 score	mAP@50	mAP@50-95
Car	0.95	0.96	0.95	0.98	0.87
Cyclist	0.93	0.91	0.91	0.95	0.83
Pedestrian	0.93	0.78	0.84	0.9	0.5
Tram	0.91	0.91	0.91	0.97	0.79
Truck	0.92	0.98	0.94	0.99	0.89
Vam	0.94	0.95	0.94	0.98	0.86

8.1.2 YOLOv12:

In our experiment for YOLOv12, we trained it for 120 epochs and the model achieved 0.95 precision, 0.91 recall, 0.95 mAP@50, and 0.82 mAP@50-95, the confusion matrix in Error! Reference source not found.34, F1 curve in Figure 35, recall curve in Error! Reference source not found., precision-recall curve in Error! Reference source not found., precision curve in Error! Reference source not found.8, result in Figure 39, and the metrics for each class all in

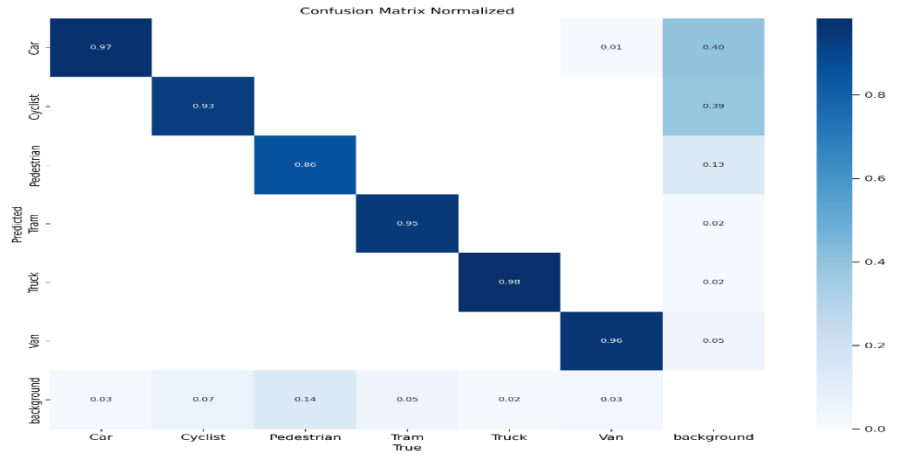


Figure (35): F1 curve for YOLOv12

Figure (36): Recall curve for YOLOv12

Figure (37): Precision-Recall curve for YOLOv12

Figure (38): Precision curve for YOLOv12

Figure (39): Training results for YOLOv12

Table (4 shows similar results to YOLOv11.

Figure (34): Confusion matrix for YOLOv12

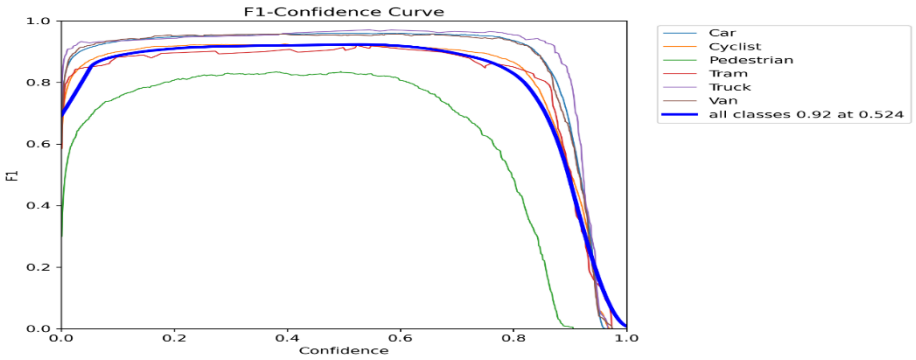


Figure (35): F1 curve for YOLOv12

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

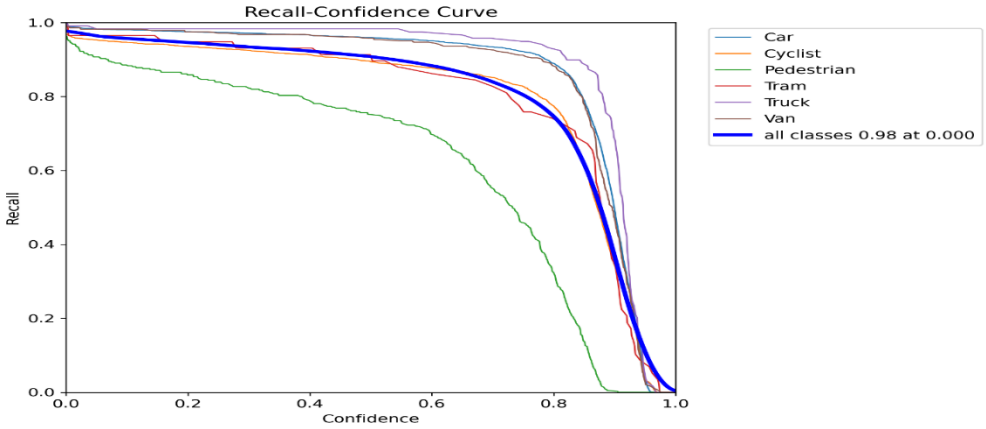


Figure (36): Recall curve for YOLOv12

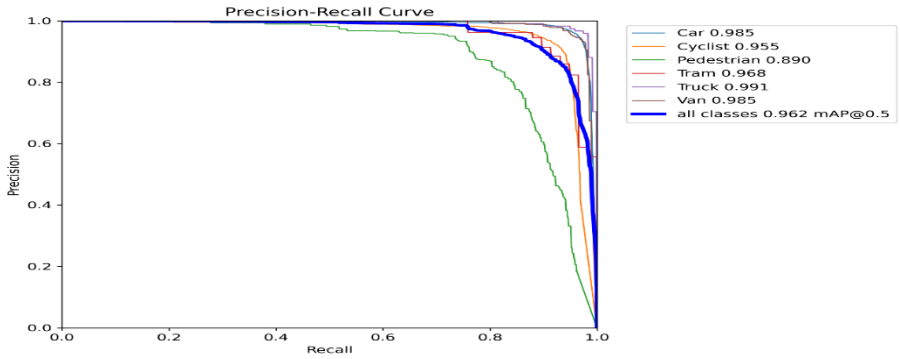


Figure (37): Precision-Recall curve for YOLOv12

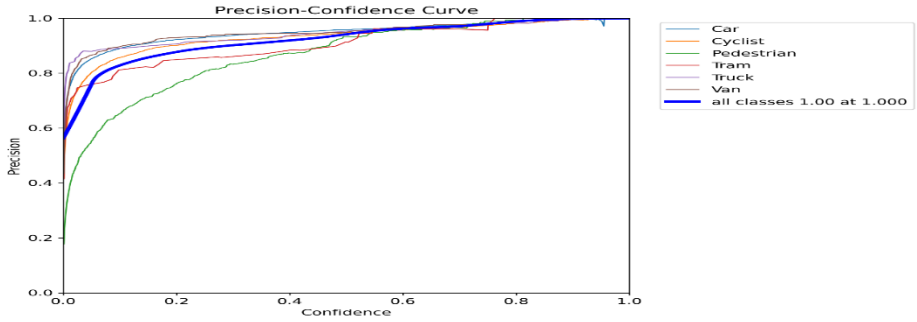


Figure (38): Precision curve for YOLOv12

International Journal of Computers and Informatics (IJCI)
Vol. (4), No. (10)
IJCI
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (4)، العدد (10)
October 2025

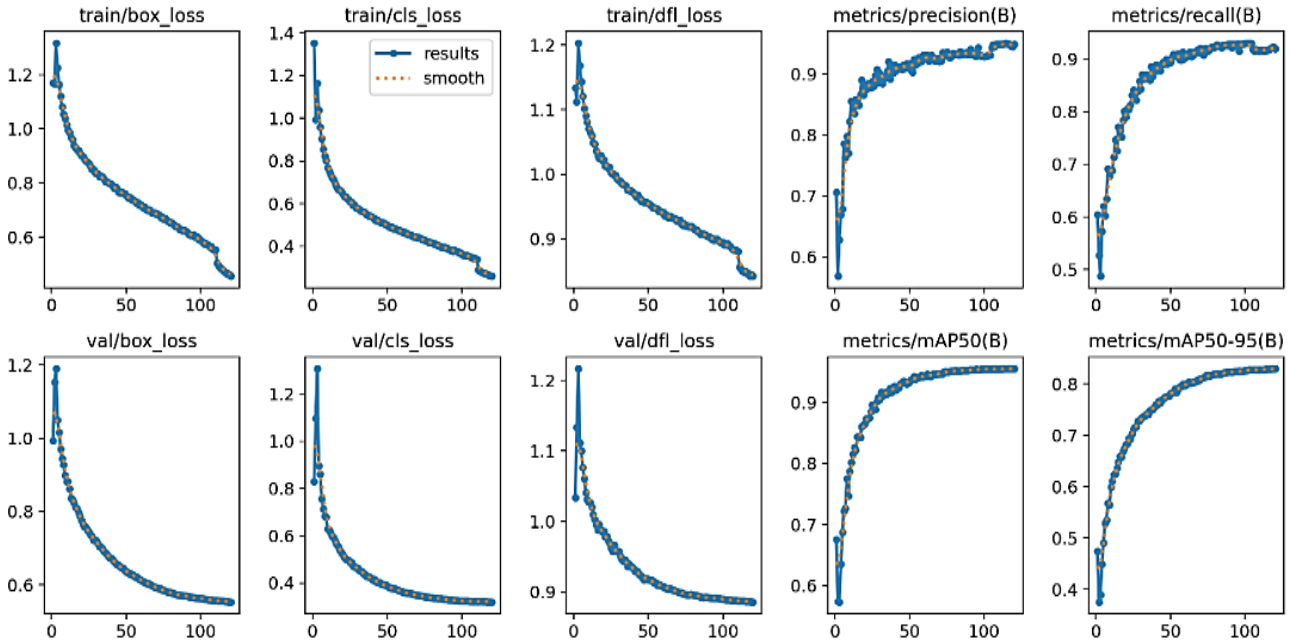


Figure (39): Training results for YOLOv12

Table (4): YOLOv12 metrics for each class

	Precision	Recall	F1 score	mAP@50	mAP@50-95
Car	0.96	0.95	0.95	0.98	0.87
Cyclist	0.95	0.89	0.91	0.95	0.83
Pedestrian	0.93	0.74	0.82	0.89	0.56
Tram	0.94	0.89	0.91	0.96	0.71
Truck	0.95	0.98	0.96	0.99	0.88
Vam	0.95	0.95	0.95	0.98	0.86

8.1.3 Explainability:

In this section, we will share a variety of pictures that showcase our explainability models in YOLOv11 and YOLOv12 on both Grad-cam++ and Eigen-cam in Figure 40, Figure 41, Error! Reference source not found.42, and Error! Reference source not found.43 while the explanation of Grad-cam++ are for majority concentrated on one point at each object with some noise, Eigen cam area of attention is larger but it shows more noise.

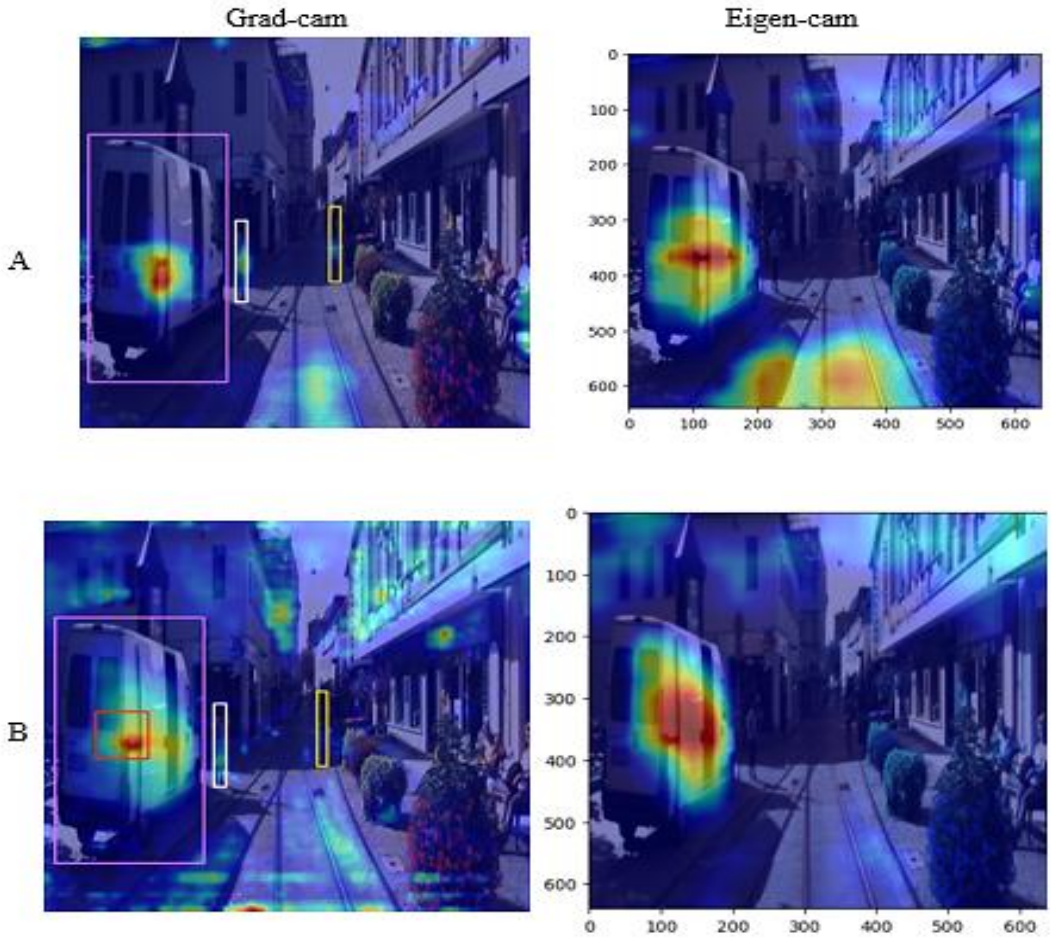


Figure (40): Picture 1 A) YOLOv11 B) YOLOv12

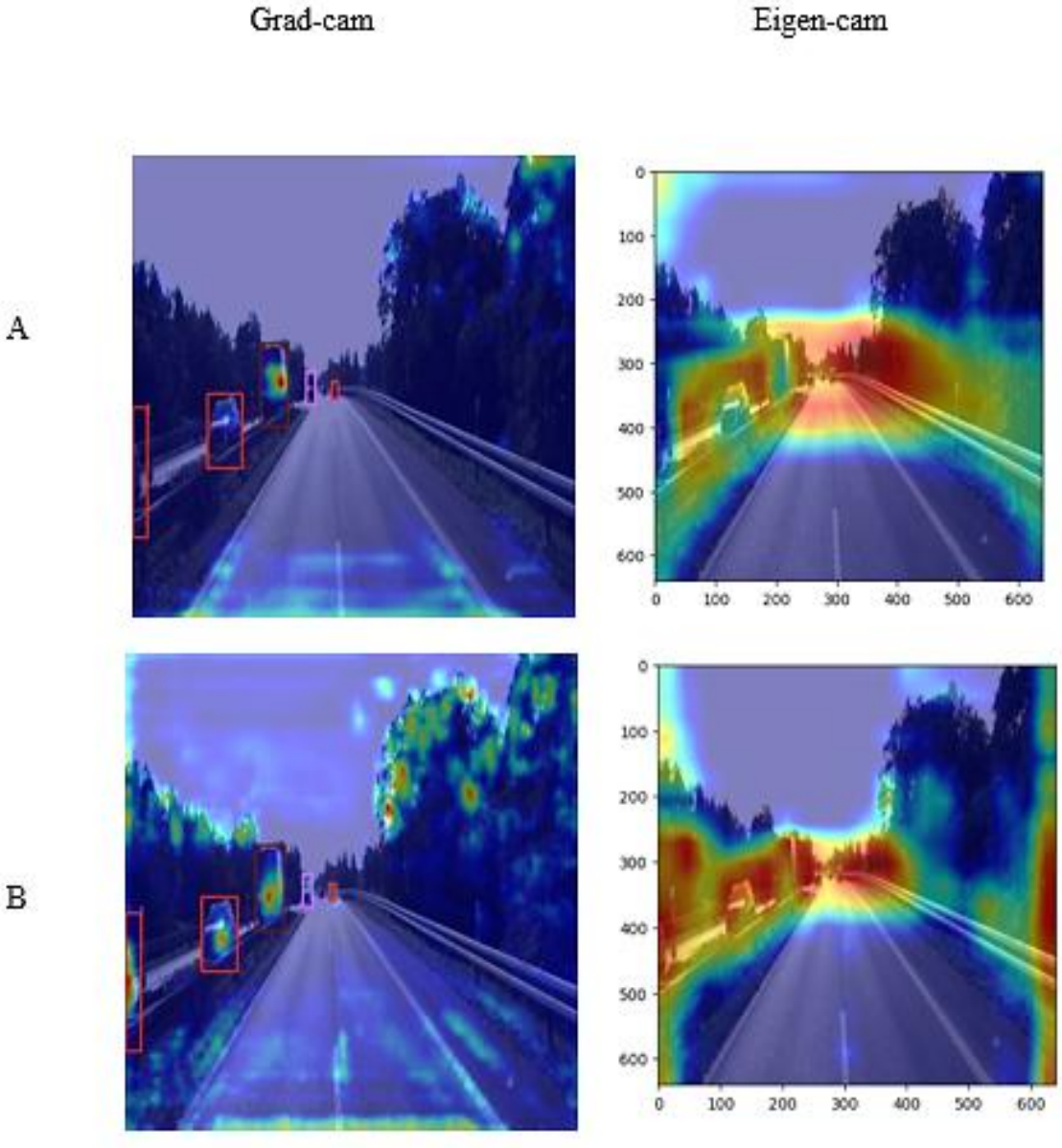


Figure (41): Picture 2 A) YOLOv11 B) YOLOv12

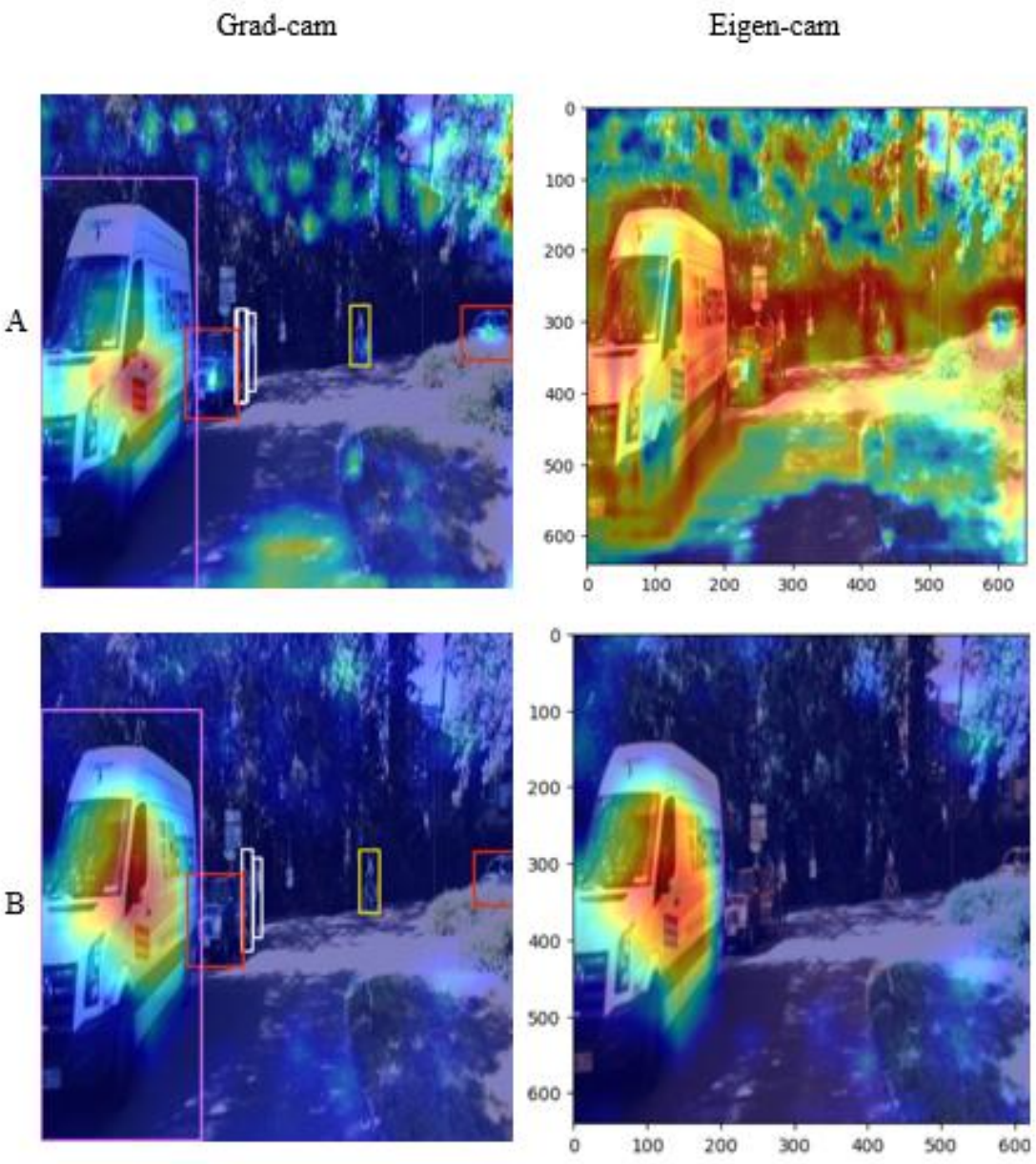


Figure (42): Picture 3 A) YOLOv11 B) YOLOv12

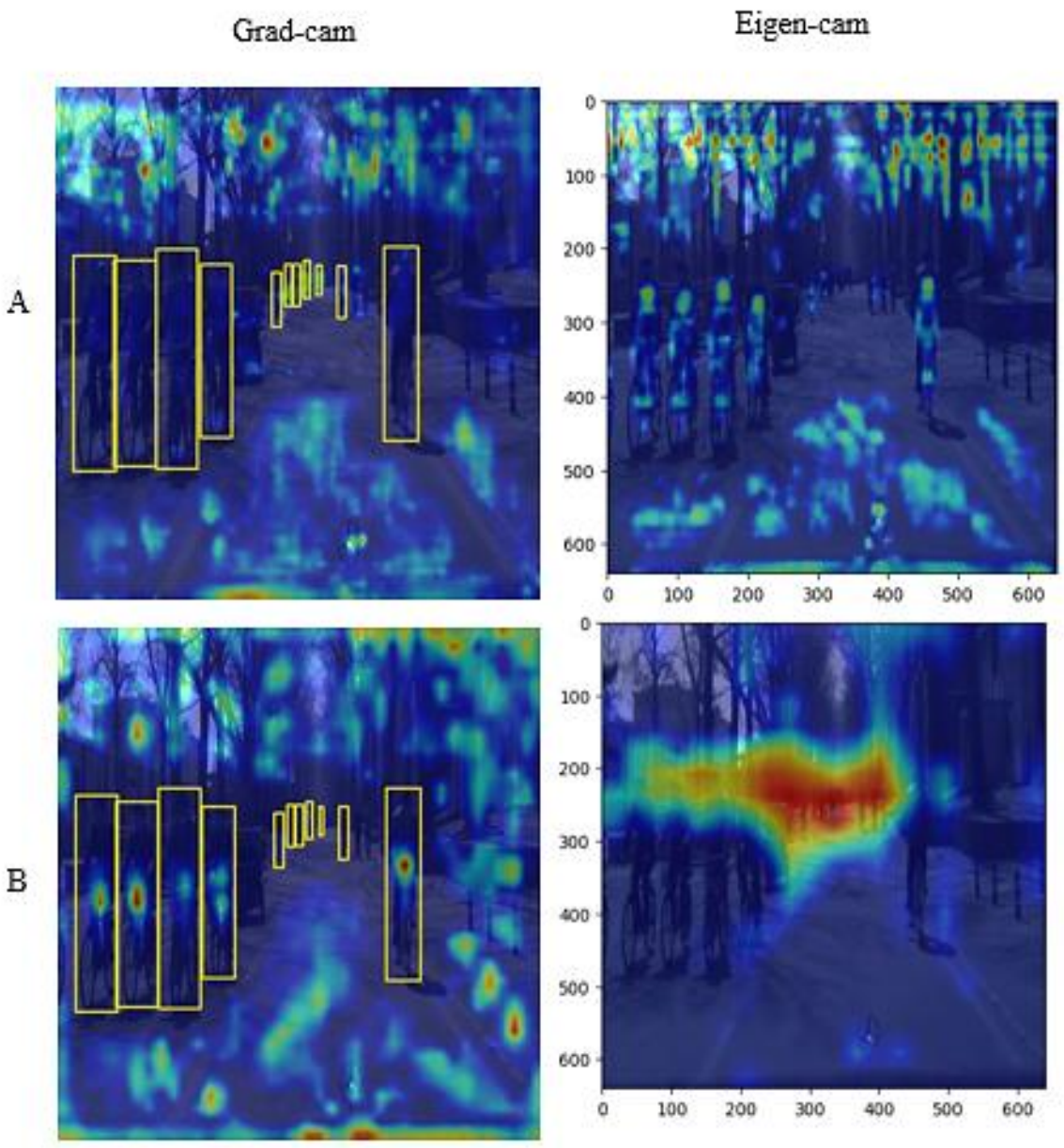


Figure (43): Picture 4 A) YOLOv11 B) YOLOv12

8.2 Comparison with Existing Work:

We will compare our YOLOv11 model with Qi wang's work in [53] and Majid Manzoor's in [54], first Qi trained an improved YOLOv11 model called BT-YOLOv11 on KITTI just like us so this comparison will be direct, his model achieved 94.4 precision, 89.9 recall, 95 mAP@50, and 77 mAP@50-75 [53], compared to our 95 precision, 91 recall, 95 mAP@50, and 82 mAP@50-95, where we had a slide edge on most metrics but same mAP@50, with Table (5 showing a comparison of mAP for each class with our YOLOv11 having better mAP for each class except tram.

Table (5): Comparison of work

	Our YOLOv11 mAP	BT-YOLOv11 mAP
Car	98	97.9
Cyclist	95	90.8
Pedestrian	90	88.2
Tram	97	99
Truck	99	97.8
Vam	98	96.5

With Majid's work in [54], YOLOv11 trained on a self-driving cars dataset achieved 0.81 mAP@50, 0.56 mAP@50-95, 0.81 F1 [54] Overall, our model's performance was better.

For YOLOv12, we couldn't find any model that was trained on autonomous vehicle object detection due to its recency when searching on Google Scholar, other work could exist in private journals.

To compare explainability, we will use Nogueira's work in [34] we used a similar picture to compare the two in Figure 44.

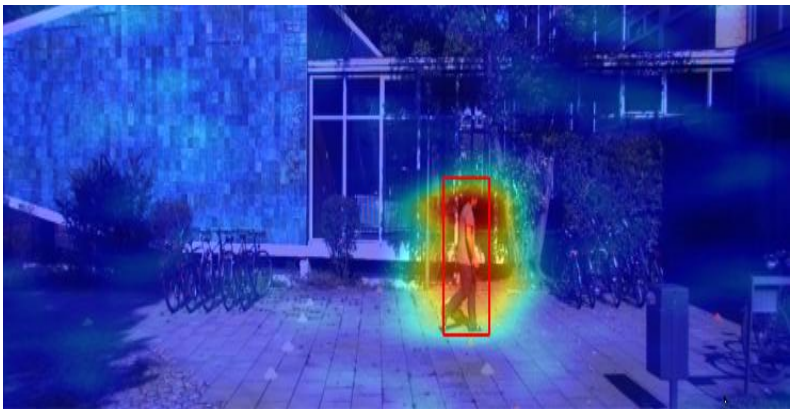


Figure (43): Nogueira explainability [34]

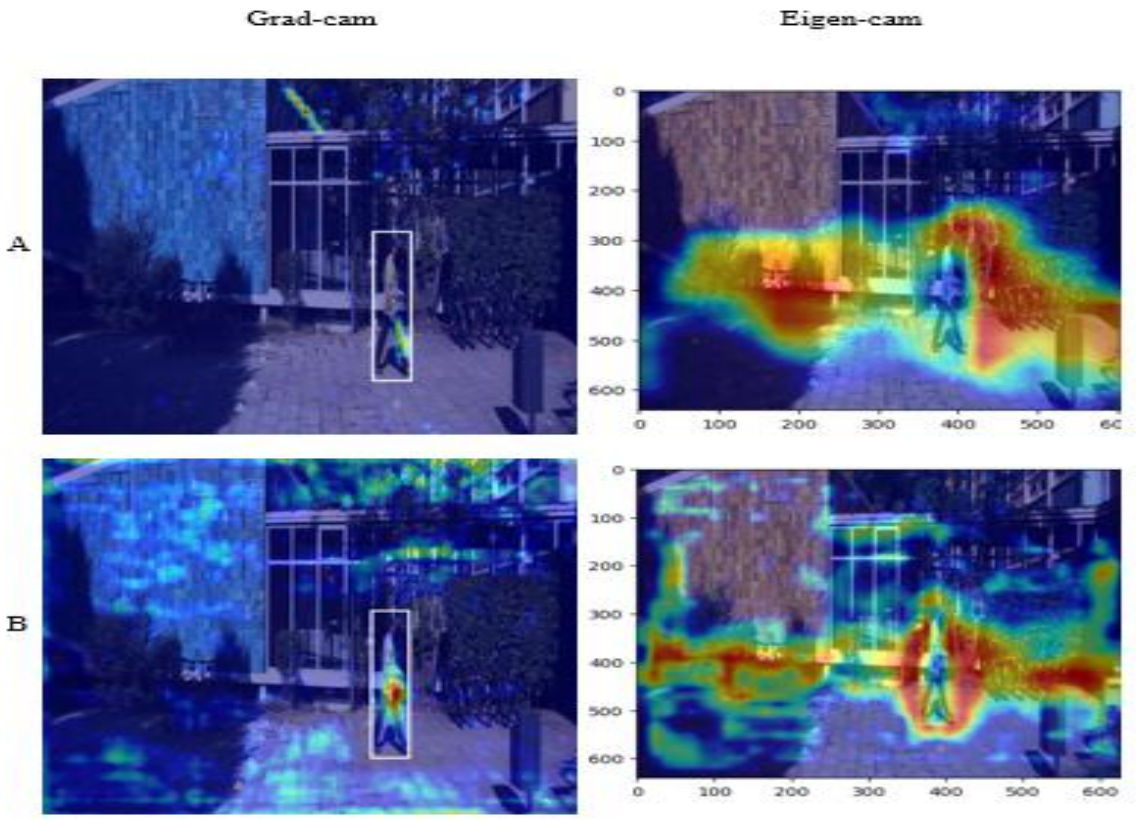


Figure (44): Our explainability A) Yolov12 B) YOLOv11 in grad-cam++ and Eigen-cam

8.3 Discussion on Findings:

Where the performance of the model was great in most classes the model is still lacking when it comes to pedestrians and needs more improvement with more pedestrian data, even though the dataset follows real-world examples with the majority being cars, it still needs to accurately detect pedestrians as the danger of not accurately detecting them every time could be catastrophic.

While observing the heat maps Grad-cam++ seemed more reliable even though the area of attention was small it concentrated on one part of the object that influenced the detection with limited noise on other unrelated regions of the image, while Eigen-cam showed big areas of focus on the big object it suffers from smaller objects and too much noise.

Overall the explanations are meaningful as you can see where the model is focusing on and what is it detecting it does increase our trust and transparency for these models even though it remains a local visual explanation it should be implemented in today's autonomous vehicles for their users to see what their car is detecting and can they trust it and feel safe while driving it or not.

9. Future work

In this project, we used CAM (class activation map) techniques such as Grad-cam++ and Eigen-cam on both YOLOv11 and YOLOv12 other explainability methods could be implemented such as layer-wise propagation (LRP), SHAP, and LIME, and even other CAM variants such as Score-cam, other datasets could be used such as Nuscnese to apply our explainability and test the generality and robustness of our model and explanations, other models could also be used where we used YOLO a one-stage detector other models that use two-stage detectors such as Faster-CNN for explaining, lastly, real-world implementation with deploying explainability into real-world vehicles could be an important next step.

10. Conclusion

In this project, we aimed to explain the YOLOv11 and YOLOv12 object detection model in an autonomous vehicle environment, we used KITTI and a cyclist dataset to train the model, achieving strong performance based on precision, recall, F1, and mean average precision (mAP), showcasing its effectiveness for detecting objects from the scene, with implementing Eigen-cam and grad-cam++ for explainability producing heatmaps that highlight regions of interest that influenced a detection to the model, showcasing a focus on object feature even when the explanation was evaluated visually it sill increase our trust, safety, and transparency the model.

The implementation was done using Python and the Ultralytics framework for YOLOv11 and YOLOv12, with pytorch-grad-cam visualizing the heatmaps. Our key contribution to this research is applying XAI techniques such as Eigen-cam and Grad-cam++ to the latest versions of YOLO version YOLOv11 and YOLOv12 in the autonomous vehicles environment, with the finding showing the impact it has on areas where safety is of the most importance, such as autonomous vehicles. With some challenges, such as measuring how good the explanation provided is, not just visually, but also using quantitative measures. Overall, this work will lay a foundation for making object detection models more interpretable and the importance of understanding how the model makes their decisions and not only how well it performs.

11. References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012. [Online]. Available: http://code.google.com/p/cuda-convnet/.

[2] A. B. Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI," *Information fusion*, Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.10045.

[3] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G. Z. Yang, "XAI-Explainable artificial intelligence," *Sci Robot*, vol. 4, no. 37, Dec. 2019, doi: 10.1126/scirobotics.aay7120.

[4] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," Feb. 01, 2019, *MDPI AG*. doi: 10.3390/s19030648.

[5] V. Hassija *et al.*, "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence," Jan. 01, 2024, *Springer*. doi: 10.1007/s12559-023-10179-8.

[6] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 11, no. 5, Sep. 2021, doi: 10.1002/widm.1424.

[7] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *ArXiv*, Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.05307.

[8] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *ArXiv*, May 2017, [Online]. Available: http://arxiv.org/abs/1705.07874.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *IEEE Xplore*, 2017, [Online]. Available: http://gradcam.cloudcv.org.

[11] V. Hassija *et al.*, "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence," Jan. 01, 2024, *Springer*. doi: 10.1007/s12559-023-10179-8.

[12] M. Reda, A. Onsy, A. Ghanbari, and A. Y. Haikal, "Path planning algorithms in the autonomous driving system: A comprehensive review," Apr. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.robot.2024.104630.

[13] K. Rana, G. Gupta, P. Vaidya, and M. Khari, "The perception systems used in fully automated vehicles: a comparative analysis," *Multimed Tools Appl*, 2023, doi: 10.1007/s11042-023-15090-w.

[14] Kamal Malik, Moolchand Sharma, Suman Deswal, Umesh Gupta, Deevyankar Agarwal, and Yahya Obaid Bakheet Al Shamsi, *Explainable Artificial Intelligence for Autonomous Vehicles: Concepts, Challenges, and Applications*. CRC press, 2024.

[15] "J3016_202104: Taxonomy and definitions for terms related to driving Automation Systems for On-Road Motor Vehicles - SAE International.," https://www.sae.org/standards/content/j3016_202104/.

[16] A. Kuznietsov, B. Gyevnar, C. Wang, S. Peters, and S. V. Albrecht, "Explainable AI for Safe and Trustworthy Autonomous Driving: A Systematic Review," *ArXiv*, Feb. 2024, doi: 10.1109/TITS.2024.3474469.

[17] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions," *IEEE Access*, vol. 12, pp. 101603–101625, 2024, doi: 10.1109/ACCESS.2024.3431437.

[18] A. Balasubramaniam and S. Pasricha, "Object Detection in Autonomous Vehicles: Status and Open Challenges," *ArXiv*, 2022.

[19] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sens (Basel)*, vol. 13, no. 1, pp. 1–23, Jan. 2021, doi: 10.3390/rs13010089.

[20] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *ArXiv*, May 2019, [Online]. Available: http://arxiv.org/abs/1905.05055

[21] J. Kim *et al.*, "Toward explainable and advisable model for self-driving cars," *Applied AI Letters*, vol. 2, no. 4, Dec. 2021, doi: 10.1002/ail2.56.

[22] J. Kim, S. Moon, A. Rohrbach, T. Darrell, and J. Canny, "Advisable Learning for Self-driving Vehicles by Internalizing Observation-to-Action Rules," *IEEE*, 2020, [Online]. Available: https://github.

[23] Y. Xu *et al.*, "Explainable Object-induced Action Decision for Autonomous Vehicles," *IEEE*, 2020.

[24] C. Li, S. H. Chan, and Y.-T. Chen, "Who Make Drivers Stop? Towards Driver-centric Risk Assessment: Risk Object Identification via Causal Inference," *IEEE*, Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.02425.

[25] H. Mankodiya, D. Jadav, R. Gupta, S. Tanwar, W. C. Hong, and R. Sharma, "OD-XAI: Explainable AI-Based Semantic Object Detection for Autonomous Vehicles," *Applied Sciences (Switzerland)*, vol. 12, no. 11, Jun. 2022, doi: 10.3390/app12115310.

[26] C.-H. Hsieh and Y.-H. Chang, "Improving DCP Haze Removal Scheme by Parameter Setting and Adaptive Gamma Correction," 2021.

[27] V. S. Saravanarajan, R. C. Chen, C. H. Hsieh, and L. S. Chen, "Improving Semantic Segmentation Under Hazy Weather for Autonomous Vehicles Using Explainable Artificial Intelligence and Adaptive Dehazing Approach," *IEEE Access*, vol. 11, pp. 38194–38207, 2023, doi: 10.1109/ACCESS.2023.3251728.

[28] M. Keser, G. Schwalbe, A. Nowzad, and A. Knoll, "Interpretable Model-Agnostic Plausibility Verification for 2D Object Detectors Using Domain-Invariant Concept Bottleneck Models," *IEEE*, 2023.

[29] J. H. Sejr, P. Schneider-Kamp, and N. Ayoub, "Surrogate Object Detection Explainer (SODEx) with YOLOv4 and LIME," *Mach Learn Knowl Extr*, vol. 3, no. 3, pp. 662–671, Sep. 2021, doi: 10.3390/make3030033.

[30] M. Alwateer, K. O. Aljuhani, A. Shaqrah, R. ElAgamy, G. Elmarhomy, and E. S. Atlam, "XAI-SALPAD: Explainable deep learning techniques for Saudi Arabia license plate automatic detection," *Alexandria Engineering Journal*, vol. 109, pp. 578–590, Dec. 2024, doi: 10.1016/j.aej.2024.09.057.

[31] M. Moradi, K. Yan, D. Colwell, M. Samwald, and R. Asgari, "Model-agnostic explainable artificial intelligence for object detection in image data," *Eng Appl Artif Intell*.

[32] Y. Li *et al.*, "A deep learning-based hybrid framework for object detection and recognition in autonomous driving," *IEEE Access*, vol. 8, pp. 194228–194239, 2020, doi: 10.1109/ACCESS.2020.3033289.

[33] M. Kuroki and T. Yamasaki, "Fast Explanation Using Shapley Value for Object Detection," *IEEE Access*, vol. 12, pp. 31047–31054, 2024, doi: 10.1109/ACCESS.2024.3369890.

[34] C. Nogueira, L. Fernandes, J. N. D. Fernandes, and J. S. Cardoso, "Explaining Bounding Boxes in Deep Object Detectors Using Post Hoc Methods for Autonomous Driving Systems," *Sensors*, vol. 24, no. 2, Jan. 2024, doi: 10.3390/s24020516.

[35] T. Ponn, T. Kröger, and F. Diermeyer, "Identification and explanation of challenging conditions for camera-based object detection of automated vehicles," *Sensors (Switzerland)*, vol. 20, no. 13, pp. 1–26, Jul. 2020, doi: 10.3390/s20133699.

[36] Tomasz Nowak, Michał R. Nowicki, Krzysztof C´wian, and Piotr Skrzypczyn´ski, "How to Improve Object Detection in a Driver Assistance System Applying Explainable Deep Learning," *IEEE*, 2019.

[37] H. A. Tahir, W. Alayed, W. U. Hassan, and A. Haider, "A Novel Hybrid XAI Solution for Autonomous Vehicles: Real-Time Interpretability through LIME–SHAP Integration," *Sensors*, vol. 24, no. 21, Nov. 2024, doi: 10.3390/s24216776.

[38] Mehdi Masmoudi, Hakim Ghazzai, Mounir Frikha, and Yehia Massoud, "Object Detection Learning Techniques for Autonomous Vehicle Applications," *IEEE*, 2019.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, [Online]. Available: https://goo.gl/bEs6Cj.

[40] Mohammed Bany Muhammad and Mohammed Yeasin, "Eigen-CAM: Class Activation Map using Principal Components," *IEEE*, 2020.

[41] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks," *ArXiv*, Oct. 2017, doi: 10.1109/WACV.2018.00097.

[42] Glenn Jocher and Jing Qiu, "Ultralytics YOLO11," https://github.com/ultralytics/ultralytics. Accessed: Dec. 09, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics.

[43] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors Latency (ms) MS COCO mAP (%)," *ArXiv*, doi: 10.0.

[44] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.17725.

[45] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *ArXiv*, Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.17725.

[46] R. Sapkota *et al.*, "YOLOv12 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series," *ArXiv*, Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.19407.

[47] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," *IEEE*, 2012.

[48] X. Li *et al.*, "A new benchmark for vision-based cyclist detection," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Aug. 2016, pp. 1028–1033. doi: 10.1109/IVS.2016.7535515.

[49] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," Apr. 2016, [Online]. Available: http://arxiv.org/abs/1604.01685.

[50] Jacob Gildenblat and contributors, "PyTorch library for CAM methods," GitHub. Accessed: May 06, 2025. [Online]. Available: https://github.com/jacobgil/pytorch-grad-cam.

[51] z1069614715, "YOLOv11 Gradcam heatmap," Github. Accessed: May 06, 2025. [Online]. Available: https://github.com/z1069614715/objectdetection_script/blob/master/yolo-gradcam/yolov11_heatmap.py.

[52] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," May 2019, [Online]. Available: http://arxiv.org/abs/1905.05055.

[53] Q. Wang and Q. L. Wang, "BT-YOLO11: Automatic Driving Road Target Detection in Complex Scenarios," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3562747.

[54] M. Manzoor *et al.*, "Obstalaneyolo: Real-Time Lane and Obstacle Detection for Autonomous Vehicles," in *International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICCWAMTIP64812.2024.10873687.

[55] T. Clement, N. Kemmerzell, M. Abdelaal, and M. Amberg, "XAIR: A Systematic Metareview of Explainable AI (XAI) Aligned to the Software Development Process," Mar. 01, 2023, *MDPI*. doi: 10.3390/make5010006.

[56] Tekedra Mawakana and Dmitri Dolgov, "Doubling down on Waymo One," Waymo. Accessed: May 08, 2025. [Online]. Available: https://waymo.com/blog/2023/07/doubling-down-on-waymo-on.