



Seminararbeit

Fahrassistenzsysteme auf Basis von maschinellem Lernen

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Luis Bros

April 2025

Matrikelnummer

6896550

Kurs

TINF22C

Dozentin

Annika Liebgott

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Aus den benutzten Quellen, direkt oder indirekt, übernommene Gedanken habe ich als solche kenntlich gemacht.

Diese Arbeit wurde bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ort, Datum

Unterschrift

Abkürzungsverzeichnis

ADAS Advanced Driver Assistance System

SAE Society of Automotive Engineers

ACC Adaptive Cruise Control

ABS Anti-Blockier-System

CNN Convolutional Neural Network

YOLO You Only Look Once

IoU Intersection over Union

mAP mean Average Precision

DNN Deep Neural Network

RPN Region Proposal Network

RoI Regions of Interest

Inhaltsverzeichnis

1. Einleitung	1
2. Optische Objekterkennung und -klassifizierung	2
2.1. Evaluationsmetriken	2
2.2. Bounding-Box-Regression	2
2.3. Klassifikation	3
2.4. One-Stage-Modelle	4
2.4.1. YOLO	4
2.4.2. RetinaNet	5
2.5. Two-Stage-Modelle	6
3. Zusammenfassung, Reflexion & Ausblick	7

1. Einleitung

Fahrerassistenzsysteme (*eng. Advanced Driver Assistance System (ADAS)*) sind in der Automobilindustrie mittlerweile weit verbreitet und zielen darauf ab, die Fahrsicherheit zu erhöhen und den Fahrkomfort zu verbessern. Sie unterstützen den Fahrer bei Ausführung der Fahraufgabe und können in verschiedenen Situationen intervenieren, um Unfälle zu vermeiden. [1]

Die verschiedenen Funktionen von Fahrerassistenzsystemen werden in verschiedene Level unterteilt, die sich durch den Grad der Automatisierung unterscheiden. Die Einordnung erfolgt in Anlehnung an die SAE J3016 Norm, die sechs Automatisierungsstufen definiert [1]:

- **Level 0:** Keine Automatisierung; Bieten lediglich Informationen, wie z.B. Parksensoren
- **Level 1:** Fahrassistenz; Der Fahrer übernimmt die Kontrolle über das Fahrzeug, aber das System unterstützt ihn, wie z.B. Anti-Blockier-System (ABS), Adaptive Cruise Control (ACC)
- **Level 2:** Teilautomatisierung; Das System übernimmt die Kontrolle über das Fahrzeug, aber der Fahrer muss bereit sein, die Kontrolle zu übernehmen, wie z.B. Highway Assist (ACC, Spurhalteassistent und Spurwechselassistent vereint)
- **Level 3:** Bedingte Automatisierung; Das System übernimmt die Kontrolle über das Fahrzeug, aber der Fahrer muss bereit sein, die Kontrolle zu übernehmen, wenn das System ihn dazu auffordert, wie z.B. „Mercedes-Benz Drive Pilot“ [2]
- **Level 4:** Hochautomatisierung; Das System übernimmt die Kontrolle über das Fahrzeug und der Fahrer muss nicht ständig bereit sein, die Kontrolle zu übernehmen, wie z.B. Waymo [3]
- **Level 5:** Vollautomatisierung; Das Fahrzeug ist vollautonom und benötigt keinen Fahrer und besitzt oft auch keine Pedale oder Lenkrad

In der vorliegenden Arbeit wird der Fokus auf die visuelle Objekterkennung gelegt, da ein signifikanter Anteil der Funktionen von Fahrerassistenzsystemen (insbesondere der Kategorien Level 0 bis 2) auf der Objekterkennung basiert. Dies passiert durch Sensoren wie Kameras, Radar und LiDAR. Diese Sensoren erfassen Informationen über die Umgebung des Fahrzeugs in Form von 2D oder 3D Representationen. So wird es dem System ermöglicht, Objekte wie andere Fahrzeuge, Fußgänger, Radfahrer und Verkehrszeichen zu erkennen und zu klassifizieren. [1]

2. Optische Objekterkennung und -klassifizierung

In modernen Fahrzeugen kommen hauptsächlich Kameras als Sensoren zum Einsatz, um die Umgebung des Fahrzeugs zu erfassen. Die so erstellten Aufnahmen enthalten Objekte, die anschließend erkannt und klassifiziert werden, um die Fahraufgabe zu unterstützen. Die Erkennung und Klassifizierung wird bei maschinellen Lernverfahren oft durch die Verwendung von Convolutional Neural Networks (CNNs) realisiert. CNNs für die Objekterkennung werden in zwei Gruppen unterteilt: One-Stage- und Two-Stage-Modelle. Die One-Stage-Modelle arbeiten mit einem einzigen CNN und bestimmen in einem Durchlauf die Wahrscheinlichkeiten einer Klasse sowie die Koordinaten der Bounding Box. [4] Im Gegensatz dazu schlagen Two-Stage-Modelle erst eine Reihe von Regions of Interest (RoI) durch ein Region Proposal Network (RPN) vor, die im zweiten Schritt an einen Objektklassifizierer gesendet werden. One-Stage-Modelle sind in der Regel schneller, weisen jedoch in der Regel eine geringere Genauigkeit auf. [5]

2.1. Evaluationsmetriken

Die mean Average Precision (mAP) ist eine Metrik, die häufig zur Bewertung der Leistung von Objekterkennungssystemen verwendet wird. Sie ermöglicht eine umfassende Bewertung der Genauigkeit und Effizienz eines Modells bei der Erkennung und Lokalisierung von Objekten in Bildern. Zur Berechnung der mAP sind zunächst die Precision und der Recall erforderlich. Die Precision bezeichnet den Anteil der korrekt erkannten Objekte an allen erkannten Objekten. Der Recall ist der Anteil der korrekt erkannten Objekte an allen tatsächlich vorhandenen Objekten. Die Average Precision wird für jede Klasse berechnet und ist die Fläche unter der Precision-Recall-Kurve. Die daraus abgeleitete mAP stellt demnach den Durchschnitt der Average Precision über alle Klassen dar. [6]

2.2. Bounding-Box-Regression

Das Bestimmen der Position eines Objekts im Raum stellt ein Bounding-Box-Regression-Problem dar. Eine Bounding Box ist eine rechteckige Region, die um ein Objekt gefasst ist. Die Bounding Boxes bestehen aus vier Werten: den X- und Y-Koordinaten der linken oberen Ecke sowie der Breite und Höhe der Box. Ziel der Bounding-Box-Regression ist die Vorhersage der Koordinaten eines Objekts. [7]

Dabei wird dem Modell ein Bild mit einer annotierten Bounding Box – der Ground Truth – gezeigt und das Modell gibt eine eigene Vorhersage ab, wo sich das Objekt

befinden könnte. Die Abweichung zwischen der Vorhersage und der Ground Truth wird anschließend mit einer Loss-Funktion berechnet. Das Modell wird trainiert, um den Fehler zu minimieren. [7]

Die Loss-Funktion für die Bounding-Box-Regression kann auf verschiedene Arten berechnet werden. Eine Möglichkeit ist der Mean Absolute Error, auch L1 Loss genannt. Er berechnet den absoluten Fehler zwischen der Vorhersage und der Ground Truth. Ein alternativer Ansatz ist der Mean Squared Error, auch L2 Loss genannt. Dieser Ansatz berechnet den quadratischen Fehler zwischen der Vorhersage und der Ground Truth. [8] Der beliebteste Ansatz ist der Intersection over Union (IoU) Loss. Er berechnet die Überschneidung der vorhergesagten Fläche mit der Ground Truth Bounding Box. Er zielt dabei auf eine möglichst hohe Überschneidung, also ein hohen IoU-Wert ab. [7]

Die Formel für das Ähnlichkeitsmaß IoU ist wie folgt definiert:

$$\text{IoU} = \frac{|B \cap B_{\text{gt}}|}{|B \cup B_{\text{gt}}|}$$

Der IoU Loss wird dann wie folgt ausgedrückt:

$$\text{IoU Loss} = 1 - \text{IoU}$$

Hierbei ist $|B \cap B_{\text{gt}}|$ die Überlappung der Flächen der vorhergesagten Bounding Box B und der Ground Truth Bounding Box B_{gt} . $|B \cup B_{\text{gt}}|$ ist die kombinierte Fläche der beiden Bounding Boxes. [7]

Mit dieser Loss-Funktion kann die Bounding-Box-Regression mit einem Stochastic Gradient Descent Algorithmus durchgeführt werden. Bei fehlender Überschneidung der Bounding Boxes ist ein Funktionieren des IoU Loss jedoch nicht möglich. Um dies zu beheben, ist die Hinzufügung eines Penalty Terms erforderlich, der die minimale normalisierte Distanz zweier Bounding Boxes anstrebt. Am besten lässt sich dies mit dem Distance-IoU Loss (DIOU) erreichen. Die Verwendung des DIOU Loss beim Training von YOLOv3 auf den PASCAL VOC Datensatz hat gezeigt, dass die Average Precision um 3,29% gegenüber dem normalem IoU verbessert werden konnte. Dies ist ein beachtlicher Wert. [7]

2.3. Klassifikation

Für die Klassifikation der Objekte in den erkannten Bounding Boxes existieren verschiedene Methoden. Zu den bekannten Methoden zählen Gradient Boosting Machi-

nes, Naive Bayes, Support Vector Machines, Logic Regression, Random Forests und CNNs. [9] Bei den CNN One-Stage-Modellen wird die Klassifikation im selben Netz durchgeführt wie auch die Bounding-Box-Regression. Bei Two-Stage-Modellen wird die Klassifikation in einem separaten Netz durchgeführt. [4] In der Praxis finden jedoch in der Regel tiefe neuronale Netzwerk-Ansätze Anwendung, da sie eine hohe Anpassungsfähigkeit aufweisen und sich für die meisten Datensätze und Anwendungsgebiete eignen. [9] Häufig wird der „Adam“-Optimizer eingesetzt, der die Gewichte während des Trainings mit adaptiver Lernrate anpasst, um die Loss-Funktion zu minimieren. Für die Hidden Layers wird eine nichtlineare Aktivierungsfunktion wie „ReLU“ verwendet und für die Output Layers die „Sigmoid“ Aktivierungsfunktion. [10], [9]

2.4. One-Stage-Modelle

2.4.1. YOLO

Ein Beispiel für ein One-Stage-Modell ist You Only Look Once (YOLO). Es handelt sich um ein schnelles und effizientes Modell, das die Objekterkennung und -klassifizierung in dem selben CNN durchführt. Das Bild wird in ein Gitter unterteilt und jede Gitterzelle ist für die Vorhersage einer Bounding Box und einer Klassenwahrscheinlichkeit verantwortlich. Das Modell wird heutzutage kontinuierlich weiterentwickelt und entspricht dem aktuellen Stand der Technik. [11]

Die YOLO-Produktfamilie ist seit dem 18. Februar 2025 in der neuesten Version 12 verfügbar. Angepasst an das Beispiel der ADAS Anwendungen wurde YOLO in einem wissenschaftlichen Benchmark auf das Traffic Sign Dataset von Radu Opera trainiert. YOLOv12x (x steht dabei für extra large) erzielte eine mAP50 von 88,9%. YOLOv8x erreichte eine mAP50 von 87,4%. [12]

Bei der praktischen Anwendung von YOLO ist es jedoch unerlässlich, die Balance zwischen Schnelligkeit, Genauigkeit und Rechenleistung zu finden. Die neuesten Modelle mit sehr hoher Genauigkeit für die Erkennung kleiner Objekte sind jedoch auch sehr rechenintensiv. Dies kann die Eignung für den Einsatz in Fahrzeugen einschränken. Für den Einsatz in ADAS Anwendungen werden daher Modelle empfohlen, die eine optimale Balance zwischen Schnelligkeit, Genauigkeit und Rechenleistung aufweisen. Für diese Anwendungen werden die Modelle YOLOv10 und YOLOv11 nano, small und medium empfohlen. [12]

Bei einem wissenschaftlichen Benchmark mit YOLOv10 nano, small und medium auf das MS COCO Dataset konnte eine Latenz pro Bild von unter 5ms erreicht werden. Die Average Precision lag dabei konstant über 40%. Diese Werte wurden mit dem

Einsatz einer High-End NVIDIA T4 GPU erreicht. [13] Obwohl der Trend dahin geht, im Auto immer mehr Rechenleistung zu verbauen, ist es dennoch wahrscheinlich, dass in Echt-Welt ADAS Anwendungen diese Latenzwerte nicht erreicht werden können. Sie geben jedoch eine gute Orientierung in welcher Größenordnung sich die Latenz bewegen könnte.

Allerdings sind auch Nachteile in Bezug auf Genauigkeit und Schnelligkeit zu berücksichtigen. Das größte Problem von YOLO und anderen One-Stage-Modellen ist, dass kleine Objekte, die in Gruppen auftauchen, nicht gut klassifiziert werden können. [11] Um das Trainingsset auch für kleine Objekte zu erweitern und die Robustheit des Modells zu erhöhen, wird Data Augmentation eingesetzt. Hierzu werden strategische Stellen des Bildes in verschiedenen Auflösungen vergrößert und daraufhin expliziert die vergrößerten Merkmale trainiert. [14] Wenn die Verbesserung des Trainingssets durch Data Augmentation nicht möglich ist, empfiehlt es sich, die large und extra-large YOLO-Modelle zu verwenden, um eine bessere Genauigkeit zu erzielen. [12]

Ein weiteres Problem bei ADAS Anwendungen ist die eingeschränkte Datenlage von Bildern in schlechten Wetterlagen. Es sind viele Datensätze für gute Wetterlagen öffentlich verfügbar, allerdings in Bedingungen wie Starkregen oder Nebel gibt es keine passenden Daten. Um das Problem der Datensammlung zu umgehen, empfiehlt es sich, ein Denoising Deep Neural Network (DNN) zu verwenden, um die schlechten Bilder vorzuverarbeiten. Das DNN wird auf den Bildern im schlechten Wetter trainiert und kann anschließend die Bilder in gute Wetterlagen umwandeln. Im Anschluss können die umgewandelten Bilder zur Objekterkennung an ein normales YOLO Modell weitergegeben werden. Um Bilder in schlechten Wetterlagen zu erstellen wurden verschiedene Data Augmentation Techniken auf dem KITTI Datensatz angewendet. KITTI ist ein bestehender Datensatz für ADAS. Mit dem Denoising DNN konnte auf diesem Datensatz eine Verbesserung der mAP mit dem YOLOv8n Modell von 4% auf 70% gezeigt werden. [15]

2.4.2. RetinaNet

Ein weiteres häufig eingesetztes One-Stage-Modell ist RetinaNet. RetinaNet ist ein einstufiges Objekterkennungsmodell, das eine Focal Loss-Funktion verwendet, um das Ungleichgewicht der Klassen während des Trainings zu berücksichtigen. Der Focal Loss ist ein dynamisch skalierbarer Cross-Entropy Loss, der den Verlust für gut klassifizierte Beispiele herabsetzt und den Verlust für schwer zu klassifizierende Beispiele erhöht. Diese Funktion hilft dem Modell, sich auf schwierige Beispiele zu konzentrieren und die Leistung bei der Erkennung seltener Klassen zu verbessern. RetinaNet ist ein einziges, einheitliches Netz, das aus einem Backbone-Netz und zwei aufgabenspezifischen

Teilnetzen besteht. Das Backbone-Netz ist für die Berechnung einer Faltungsmerkmalskarte über das gesamte Eingabebild verantwortlich und ist ein Standard CNN. Das erste Teilnetz führt eine Objektklassifizierung mittels CNNs am Ausgang des Backbone durch, während das zweite Teilnetz eine Bounding-Box-Regression mittels CNNs durchführt. [16]

RetinaNet soll durch den Focal Loss so akkurat sein wie Two-Stage-Modelle, aber dennoch den Vorteil der Schnelligkeit von One-Stage-Modellen behalten. Deswegen eignet es sich gut für Aufgaben im Auto. ADAS Anwendungen wie die Erkennung von dynamischen Informationsschildern auf Straßen sind mögliche Anwendungsbereiche von RetinaNet. [4]

2.5. Two-Stage-Modelle

Wie der Name der Two-Stage-Modelle bereits andeutet, ist der Erkennungsprozess in zwei klar definierte Stufen unterteilt [17]:

1. Region Proposal (Erste Stufe): In dieser Stufe werden mit Hilfe eines CNN Backbone-Netzwerk hierarchische Eigenschaften aus dem Eingabebild extrahiert. Basierend auf diesen Eigenschaften wird dann mittels RPN potenzielle Bereiche im Bild identifiziert, die möglicherweise Objekte enthalten. Dieser Schritt beantwortet die Frage „Wo könnten Objekte sein?“ und generiert sogenannte Region Proposals oder RoI.
2. Klassifikation und Verfeinerung (Zweite Stufe): In dieser Stufe werden die RoI genauer analysiert und mittels Klassifikator eingestuft. Durch den Bounding-Box-Regressor werden die genauen Positionen der Objekte in den RoI bestimmt. Diese Phase beantwortet die Frage „Was ist in den identifizierten Regionen?“. Sie liefert die endgültigen Klassifikationen und Bounding Boxen für die erkannten Objekte.

Die bekanntesten Vertreter der Two-Stage-Modelle gehören zur Familie der R-CNN (Region-based Convolutional Neural Network). Das neueste Modell der R-CNN Familie ist das Mask R-CNN. Es ist eine Erweiterung des Faster R-CNN und fügt eine zusätzliche Segmentierungsmaske für die Objekte in den RoI hinzu. Diese Erweiterung ermöglicht eine präzisere Segmentierung der Objekte im Bild. [18] Das Faster R-CNN fügt ein RPN hinzu. Das RPN ist ein „fully convolutional network (FCN)“ [17], das die Region Proposals generiert. Es nutzt eine Sliding-Window-Technik, um verschiedene Ankerboxen über das Bild zu schieben und die Wahrscheinlichkeit zu berechnen, dass sich in jeder Box ein Objekt befindet. Das RPN wird mit Back-Propagation und dem

Stochastic Gradient Descent Algorithmus trainiert. [17]

Das Training von Faster R-CNN auf den PASCAL VOC 2007 und 2012 Datensätzen und das anschließende Fine-Tuning mit dem MS COCO Datensatz resultieren in einer mAP von 78,8% auf dem PASCAL VOC 2007 Datensatz. Dieser Wert ist als sehr gut einzustufen, allerdings ist das Modell auch sehr rechenintensiv und benötigt viel Rechenleistung. Pro verarbeitetem Bild werden ca. 200ms benötigt. Dies erlaubt einem Objekterkennungssystem nur etwa fünf Bilder pro Sekunde zu verarbeiten. [17] Für moderne ADAS ist die Verarbeitungszeit pro Bild jedoch um Größenordnungen zu hoch. Für Anwendungen, die kritische Entscheidungen erfordern, ist eine niedrigere Verarbeitungszeit daher unerlässlich.

3. Zusammenfassung, Reflexion & Ausblick

Für ADAS Anwendungen werden One-Stage-Modelle bevorzugt, da sie schneller sind und auf die begrenzte Rechenleistung der Fahrzeuge passen. [4] Gerade das beliebte YOLO Modell konnte zeigen, dass es um Vielfaches schneller als Two-Stage-Modelle wie Faster R-CNN ist. [17]

Folgende Probleme in der Objekterkennung wurden in dieser Arbeit behandelt und Mögliche Lösungen vorgestellt:

- **Kleine Objekte:** Hier wurde eine Data Augmentation Technik vorgestellt, die die Merkmale von verschiedenen vergrößerten Auflösungen kombiniert, um kleine Objekte besser zu erkennen. Auch empfiehlt es sich, größere Modelle zu verwenden.
- **Verdeckungen bei schlechten Wetterlagen:** Hier wurde eine Data Augmentation Technik vorgestellt, die Regen- und Nebel effekte erzeugt. Diese können dann mit einem Denoising DNN entfernt werden.
- **Echt-Zeit Performance:** Hierzu wurden One- und Two-Stage-Modellen verglichen und erkannt, dass kleine One-Stage-Modelle wie YOLO und RetinaNet für Echt-Zeit Anwendungen besser geeignet sind.

Weitere Untersuchungen auf diversere Datensätze sind notwendig, um die Robustheit weiter zu steigern. Es müssen mehr Datensätze in schwirigen Wetterlagen aus der Echten Welt erhoben werden, um diese Systeme universell und ortsunabhängig anbieten zu können.

Zusätzlich sollten zukünftige Untersuchungen auf die Integration von Multi-Sensor-

Datenfusion abzielen, um die Genauigkeit und Robustheit der Objekterkennung zu verbessern. Die Kombination von Kameradaten mit LiDAR, Radar oder anderen Sensordaten könnte helfen, die Einschränkungen einzelner Sensoren zu überwinden.

Ein weiterer wichtiger Aspekt ist die Untersuchung von energieeffizienten Modellen, die speziell für den Einsatz in ressourcenbeschränkten Umgebungen wie Fahrzeugen optimiert sind. Hier könnten Techniken wie Modellkompression, Quantisierung und prädiktive Codierung eine Rolle spielen.

Schließlich ist die Entwicklung von Modellen, die besser mit seltenen oder unerwarteten Szenarien umgehen können, von entscheidender Bedeutung. Dies könnte durch den Einsatz von Generative Adversarial Networks (GANs) oder anderen Methoden zur Erzeugung synthetischer Daten erreicht werden, um die Modelle auf Edge Cases vorzubereiten.

Literatur

- [1] M. Galvani, “History and future of driver assistance,” *IEEE Instrumentation & Measurement Magazine*, vol. 22, no. 1, pp. 11–16, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8633345>
- [2] Drive pilot | mercedes-benz. [Online]. Available: <https://www.mercedes-benz.de/passengercars/technology/drive-pilot.html>
- [3] W. LLC, “Challenges for the evaluation of automated driving systems using current ADAS and active safety test track protocols,” 2023. [Online]. Available: <https://waymo.com/research/challenges-for-the-evaluation-of-automated-driving/>
- [4] G. De-Las-Heras, J. Sanchez-Soriano, and E. Puertas, “Advanced driver assistance systems (adas) based on machine learning techniques for the detection and transcription of variable message signs on roads,” *Sensors (Basel, Switzerland)*, vol. 21, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:237445540>
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014. [Online]. Available: <https://arxiv.org/abs/1311.2524>
- [6] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010. [Online]. Available: <https://link.springer.com/article/10.1007/s11263-009-0275-4>
- [7] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-iou loss: Faster and better learning for bounding box regression,” 2019. [Online]. Available: <https://arxiv.org/abs/1911.08287>
- [8] R. Girshick, “Fast r-cnn,” 2015. [Online]. Available: <https://arxiv.org/abs/1504.08083>
- [9] R. Ghandour, A. J. Potams, I. Boulkaibet, B. Neji, and Z. Al Barakeh, “Driver behavior classification system analysis using machine learning methods,” *Applied Sciences*, vol. 11, no. 22, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/22/10562>
- [10] A. Liebgott, “Kapitel 7 - Neuronale Netze und Deep Learning,” 2 2022. [Online]. Available: https://elearning.dhbw-stuttgart.de/moodle/pluginfile.php/847448/mod_resource/content/1/Kapitel7_DeepLearning.pdf

- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [Online]. Available: <https://ieeexplore.ieee.org/document/7780460>
- [12] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, “Yolo evolution: A comprehensive benchmark and architectural review of yolov12, yolo11, and their previous versions,” 2025. [Online]. Available: <https://arxiv.org/abs/2411.00201>
- [13] A. Wang *et al.*, “Yolov10: Real-time end-to-end object detection,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.14458>
- [14] W. Liu *et al.*, *SSD: Single Shot MultiBox Detector*. Springer International Publishing, 2016, p. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2
- [15] M. Z. Shahzad, M. A. Hanif, and M. Shafique, “Robust adas: Enhancing robustness of machine learning-based advanced driver assistance systems for adverse weather,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.02581>
- [16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2018. [Online]. Available: <https://arxiv.org/abs/1708.02002>
- [17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7485869>
- [18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988. [Online]. Available: <https://ieeexplore.ieee.org/document/8237584>